国外电子与通信教材系列

CDMA 蜂窝移动通信与网络安全

CDMA Cellular Mobile Communications and Network Security

[韩] Man Young Rhee 著

袁超伟 等译



電子工業出版社



Pearson Education
培生教育出版集团

内容简介

本书主要介绍了CDMA原理与系统，能够满足各种层次人员的需要。CDMA的网络安全是本书的特色。本书采用了大量的实例和数学运算来帮助读者理解CDMA蜂窝系统、消息加密和网络安全。全面而系统地讲述了CDMA信道操作技术，反向和前向CDMA信道，移动台呼叫处理，基站呼叫处理，单向Hash函数和消息摘要，鉴权、加密和识别，前向和反向W-CDMAwy uthp. +本书适用于从事CDMA蜂窝系统深入研究与开发的电信工程师、工程管理人员，同时对在这个领域进行教学、研究、开发的教师、学生有很好的参考价值。

版权贸易合同登记号：图字：01－2002－0000

图书在版编目（CIP）数据

CDMA蜂窝移动通信与网络安全./(韩)凯曼(Kamen,M.)著；袁超伟等译. －北京：电子工业出版社，2002.0

（国外电子与通信教材系列）

书名原文：CDMA Cellular Mobile Communications and Network Security

ISBN 7－0000－0000－0

Ⅰ. C... Ⅱ. ①... ②... Ⅲ. xxxxxxx, xxxxxxxx Ⅳ. TP000.00

中国版本图书馆CIP数据核字（2002）第000000号

责任编辑：李秦华　特约编辑：

印　刷　者：

出版发行：电子工业出版社　http://www.phei.com.cn

北京市海淀区万寿路173信箱　邮编：100036

经　　销：各地新华书店

开　　本：787×1092　1/16　　印张：24.5　字数：627.2千字

版　　次：2002年5月第1版　2002年5月第1次印刷

定　　价：00.00元（含光盘）

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077

译 者 序

20 世纪 90 年代以来，移动通信发展迅速，我国现已有移动用户 1.5 亿。码分多址（CDMA）技术以它在系统容量、业务质量、安全性和可靠性等方面的优势倍受关注，全球已有近 50 个国家采用 CDMA 技术提供电信业务。新建成的中国联通 CDMA 网一期工程总投资 240 亿元人民币，采用 IS-95A 增强型技术标准，网络规模为 1515 万户覆盖全国 31 个省市的 330 个本地网，除开通基本通话业务和各种附加功能外，还开通移动智能网和短消息网等增值业务。七城市的 1X 试验网规模总容量为 10 万户，已经和正在开展高速上网、图像下载、可视电话等新业务试验。CDMA 是一种先进的数字通信技术，经历了从窄带 CDMA（IS-95 CDMA）向宽带 CDMA 的演进，已成为第三代移动通信系统的实际应用技术。

本书全面而系统地介绍了 CDMA 原理与系统，能够满足各种层次人员的需要，CDMA 的网络安全是本书的特色。本书采用了大量的实例和数学运算来帮助读者理解 CDMA 蜂窝系统、消息加密和网络安全。

本书适用于从事 CDMA 蜂窝系统方面深入研究与开发的电信工程师、工程管理人员，同时对在这个领域进行教学、研究、开发、应用的教师和学生有很好的参考价值。

参与本书翻译工作的有袁超伟、张元雯、陈文杰、王桃荣、刘静、徐冬云、孙楠、章高男、张元馨等。同时，该书的翻译工作还得到了所在部门和相关领导的大力支持，在此一并表示感谢。受译者水平所限，错误和不妥之处在所难免，恳请读者批评指正。

前　言

本书主要讨论了 CDMA（码分多址）技术。作为未来移动通信系统前景广阔的技术，CDMA 正得到广泛的关注。

对于蜂窝系统而言，选择最恰当的接入方式是很具挑战性的工作。对于宽带服务，CDMA 是一种很有吸引力的无线接入技术。为了迎接这种挑战，必须熟悉 CDMA 数字蜂窝系统的相关技术和系统体系结构。

在过去的 6 年中，无线通信领域业已发生了巨变。本书旨在激发读者进一步探索这个极具挑战性的领域。1990 年初，由位于美国加利福尼亚圣迭戈的 QUALCOMM 股份有限公司（高通公司）先驱性地引入了关于 CDMA 扩频数字蜂窝系统概念和开创性的实现方法。该 CDMA 系统后来得到相关标准化组织的认可，已成为电信工业协会（TIA）和电子工业协会（EIA）的 IS-95 标准。

本书给出了全面的分析方法，有助于从业工程师规划和设计高效的 CDMA 蜂窝组网。本书也适合研究生学习扩频蜂窝系统的基本原理。本书中给出的大部分内容，特别是 CDMA 信道结构，均包含在 IS-95 原理和系统体系结构中。

本书中大量计算例子以数值表示，为的是让初学者更好地理解 CDMA 蜂窝系统。

下面是本书每章内容的摘要。

第 1 章概述了 CDMA 蜂窝系统的介绍性材料，简单解释了基于扩频系统的调制和多重接入技术。

第 2 章介绍了 CDMA 信道操作所需的最基本和实用的技术基础。

第 3 章和第 4 章涵盖了 CDMA 信道全面而详细的体系结构以及它们特性和功能的讨论。前向 CDMA 信道由导频、同步、寻呼和业务信道组成。导频信号是每个 CDMA 基站一直发送的不调制、直接序列扩频信号。移动台监视导频信道，以捕获前向 CDMA 信道定时并提供相关解调的相位参考。同步信道向移动台传送同步消息，为的是获得初始时间同步。寻呼信道也是经过编码、交织、扩频和调制的扩频信号，用来发送从基站到移动台的控制信息和寻呼信息。前向业务信道用来在通话过程中从基站向特定移动台发送用户数据和信令业务。除了导频信道，其他码道都经过了卷积编码、块交织、适当的 Walsh 函数正交扩频，再由导频 PN 序列正交对以 1.228 8 Mc/s 的速率扩展。

寻呼信道和前向业务信道也采用数据扰码技术。数据扰码在块交织器输出端以 19.2 kb/s进行。数据扰码通过对交织器输出字符与用户号码对应的长伪码进行模 2 加。长码是周期为 $2^{42}-1$ 的 PN 序列，用于前向 CDMA 信道（即寻呼和前向业务信道）的扰码和反向 CDMA 信道（即接入和反向业务信道）的扩频。反向 CDMA 信道由接入信道和反向业务信道组成。在反向 CDMA 信道上发送的所有数据都经过卷积编码、块交织、64 阶 Walsh 函数正交调制和长码片直接序列扩展而发送。反向业务信道使用数据脉冲随机发生器产生掩码数据 0 和 1，用来掩蔽码重复产生的冗余数据。

第 5 章和第 6 章描述的是基于高通系统包括切换过程的 CDMA 码道呼叫过程。

第 7 章给出了单向 Hash 函数和消息摘要的简明概要。无论对传统的对称算法还是公钥加密算法,单向函数是大多数协议的基本创建单元。单向 Hash 函数非常容易计算,但是极难逆运算。用于鉴权数据的 Hash 码算法是系统地排列的。

第 8 章讲述鉴权和消息保密。分析范围涉及大量由 152 比特 CDMA 蜂窝系统消息块到 18 比特 Hash 码的运算技术。移动台和基站协同操作对移动台进行鉴权。鉴权是这样一个过程,即信息在移动台和基站之间交互用于确认移动台身份。只有当能够表明移动台和基站处理的 SSD(共享加密数据)集相同的时候,鉴权过程才成功。SSD 是半永久存储在移动台存储器中的 128 比特长的共享加密数据。SSD 分为两个不同子集:SSD-A 和 SSD-B。前者用于支持鉴权过程;后者用于支持话音保密和消息加密。SSD 更新过程可以完全用 SSD-A-NEW 和 SSD-B-NEW 解释,并作为 SSD 产生输出。同时还包括信令消息加密和网络安全。

第 9 章和第 10 章介绍基于 JTC(AIR)/95 的宽带 CDMA 链路。反向和前向信息信道数据率为64 kb/s,32 kb/s 或 16 kb/s,PN 码片速率为 4.096 Mc/s,符号速率为 64 ks/s。

第 9 章讲述反向 W-CDMA 信道,它是从移动台到基站的通信链路。反向 W-CDMA 信道由接入信道和反向业务信道组成。接入信道包括反向导频信道和反向接入信道。反向业务信道包括三种不同信道:反向导频、信息和信令信道。反向业务信道上的所有数据都经过卷积编码、交织和直接序列扩展调制再发送。在第 10 章中,前向 W-CDMA 信道包括 1 路导频信道、1 路同步信道、最多 8 路寻呼信道和一定数量前向业务信道(如前向信息和信令信道)。这些码道中每一路都由适当的 Walsh 码正交扩展,然后以 4.096 Mc/s 固定码片速率的 PN 序列扩展。前向业务信道中的前向信令信道经过卷积编码、块交织、长码直接序列扩展、Walsh 函数正交扩展、以 4.096 Mc/s 固定码片速率的 PN 序列正交调制、基带滤波和 QPSK 波形发送。

本书可以作为帮助读者进行数字蜂窝系统技术方面深入研究和开发的基本教材。希望书中总结的带有完整解答的 148 个问题更有益于读者自学。在这个飞速发展的领域早期阶段写这本书确实是种很大的挑战。希望读者能发现书中存在的问题,真诚欢迎读者的反馈,这将有助于提高本书的再版质量。

目　　录

第 1 章　CDMA 蜂窝网介绍

扩频调制技术已经历了过去 40 多年来的演化。扩频技术曾经广泛用于抗干扰和多径场合以及测距和跟踪。扩频技术还被用于 CDMA，以支持在大量群体用户之间同时进行数字通信的服务。CDMA 概念可简单地解释成基于扩频通信的调制和多址接入方案。

本章概要介绍了美国圣迭戈高通公司倡导的 CDMA 数字蜂窝系统。在很多参与其中的通信公司和设备制造商（ AT&T，Motorola，North Telecom 和其他）的协作下，基于多址接入方案的数字蜂窝应用也取得了进展。CDMA 系统作为候选标准（IS-95）完全符合蜂窝通信工业协会（CTIA）要求。

典型的数字蜂窝系统有 GSM（欧洲 1990 年提出的方案）、NATDMA（北美 1990 年提出的 IS-54 方案）、PDC（日本 1990 年提出的标准方案）以及 CDMA（美国 1993 年提出的 IS-95 方案）。

1982 年 6 月，西欧提出了基于时分多址（TDMA）的 GSM 系统。GSM 能够扩展多样的电信网络（例如 ISDN），并提供了对整个欧洲大陆的兼容性。1992 年，第一个商用 GSM 系统在德国设计成功。GSM 基于频分多址和时分多址的组合。

NA-TDMA 系统和 GSM 相似，惟一差别在于该系统中仅仅存在一个公共无线接口。PDC（个人数字蜂窝）是日本提出的 TDMA 蜂窝系统，工作在 800 MHz 和 1.5 GHz。该系统在数字蜂窝网络之间提供了 9 个接口。1.5 GHz PDC 于 1994 年公开投入运营。

除了数字多址接入系统，还有 TDD 无绳电话系统，如 PHP，CT-2，DCT-900（或 CT-3）以及 DECT。TDD（时分双工）系统都是数字系统，但只使用一个载波发送和接收信息。PHP（个人便携式电话）是支持 PCS（个人通信服务）的 TDD 无线通信系统。PHP 可以用于住宅无绳电话、私有无线 PBX（专用分组交换机）、公众远程点和无线电话通信。CT-2（无绳通信系统 2）是第二代无绳电话系统，在英国由 GPT 公司开发，是第一个用于无线移动通信的 TDD 系统。CT-2 系统是一种最简单的 PCS 系统，具有不需要多信道复用的控制结构。CT-2 系统不使用信道编码，取消了切换和寻呼，因此只允许出呼叫服务。呼叫半径通常小于 200 米，允许用户占用很大带宽。DCT-900 或 CT-3（工作在 900 MHz 的数字无绳电话）由瑞典爱立信公司在 1988 年开发，作为 CT-2 的升级版，有时又被称为 CT-3。英国和瑞典允许使用 CT-2 和 DCT-900 系统，直到 DECT 的出现。DECT（数字欧洲无绳通信）像 CT-2 和 DCT-900 一样被采用成为无绳电话欧洲标准，但它具有改进的资源用于处理数据和语音传输。CDMA 发展始于 1989 年早期，在 NA-TDMA 标准（IS-54）创立之后。1989 年 11 月进行了 CDMA 可行性测试。EIA 的 CDMA IS-95 中间标准在 1992 年 12 月发布。CDMA 系统可以使用双模用户单元提供与模拟系统的兼容性。但本书仅限于讨论 CDMA 数字模式。

1.1　CDMA 蜂窝覆盖范围

对 CDMA 蜂窝系统，服务区被划分为图 1.1 所示的六边形蜂窝小区。每个小区包含一个基站，在话音编码和解码之前，基站与 MTSO（移动电话交换局）连接。每个小区中，在基站和小区内每个移动台之间存在两条由前向和反向 CDMA 信道组成的链路。前向 CDMA 信道是指从基站到

小区内移动台的前向链路。反向 CDMA 信道是指从移动台到基站的反向链路。

CDMA 重复使用小区无线电频率资源并高效地控制系统容量，因为 CDMA 系统天生具有出色的抗干扰能力。

前向 CDMA 信道包括一个或多个码道，这些码道在某一 CDMA 指定频率传输，并使用特定的导频 PN 偏移。每个基站使用一个导频 PN 序列的时间偏移（称为伪随机扩频序列）来标识每个前向 CDMA 信道。CDMA 蜂窝系统中，时间偏移可以重复使用。

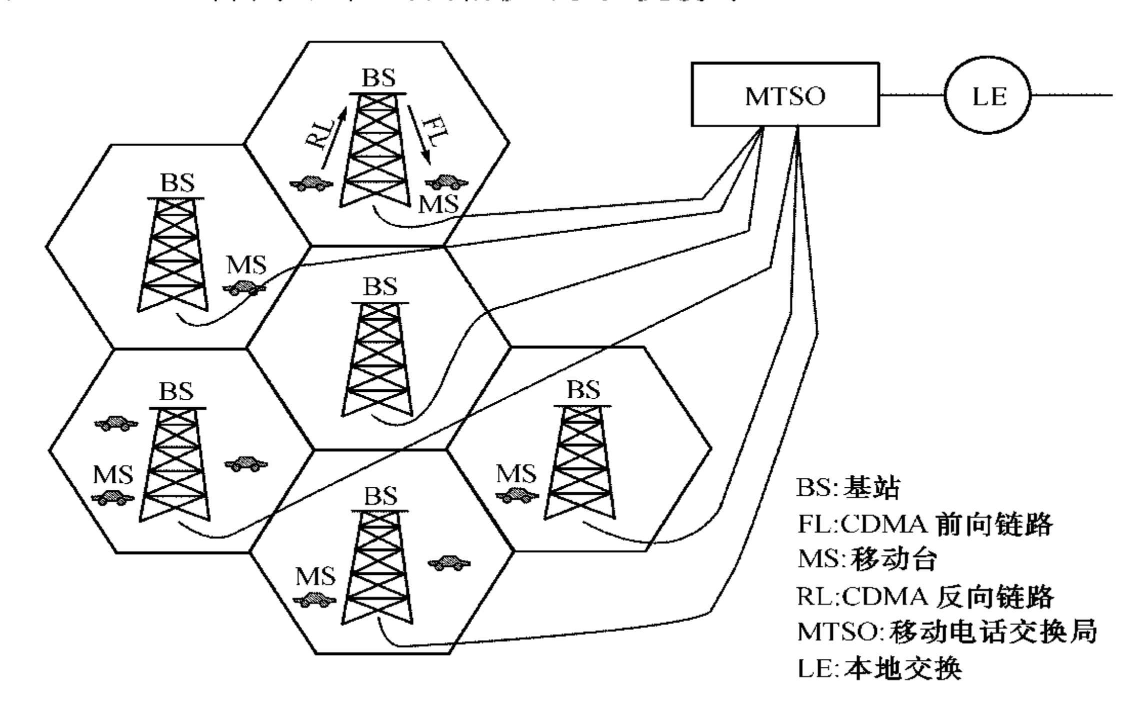


图 1.1　六边形蜂窝小区覆盖范围 CDMA 前向/反向小区链路几何图

前向 CDMA 信道上每个码道由适当的 Walsh 函数正交扩展，以使得所有码道之间正交，再由导频 PN 序列正交对（即相互正交的相位）扩频，以便使用 QPSK（正交相移键控）调制波形传输。

反向 CDMA 信道由接入信道和反向业务信道组成。接入信道用于短信令消息交换，能提供呼叫来源、寻呼响应、指令和注册。反向业务信道用于从单个移动台向单个或多个基站传输用户信令和业务。需要在反向 CDMA 信道传输的所有数据都经过卷积编码，以便纠错、块交织，从而防止突发错误，并引入接入冗余来提高系统性能，经过 64 阶 Walsh 函数调制来提供正交信道化，以及由长码进行直接序列扩频来获得发送之前有限的加密。

1.2　CDMA 信道的结构分层

图 1.2 和图 1.3 分别显示了前向/反向 CDMA 信道的完整结构。前向 CDMA 链路包括导频信道、同步信道、寻呼信道和一定数量的前向业务信道。典型的前向 CDMA 信道包含了 64 路码道可供使用。在这 64 条码道中，前向 CDMA 链路包括 1 路导频信道、1 路同步信道、7 路寻呼信道和 55 路前向业务信道。

导频信道用于传送导频信息，由 CDMA 基站连续不断地发送一种不调制的直接序列扩频信号。移动台监视导频信道以捕获前向 CDMA 信道定时并提供相关解调的相位参考。导频码道总是分配编号为 0（W_0）的码道。

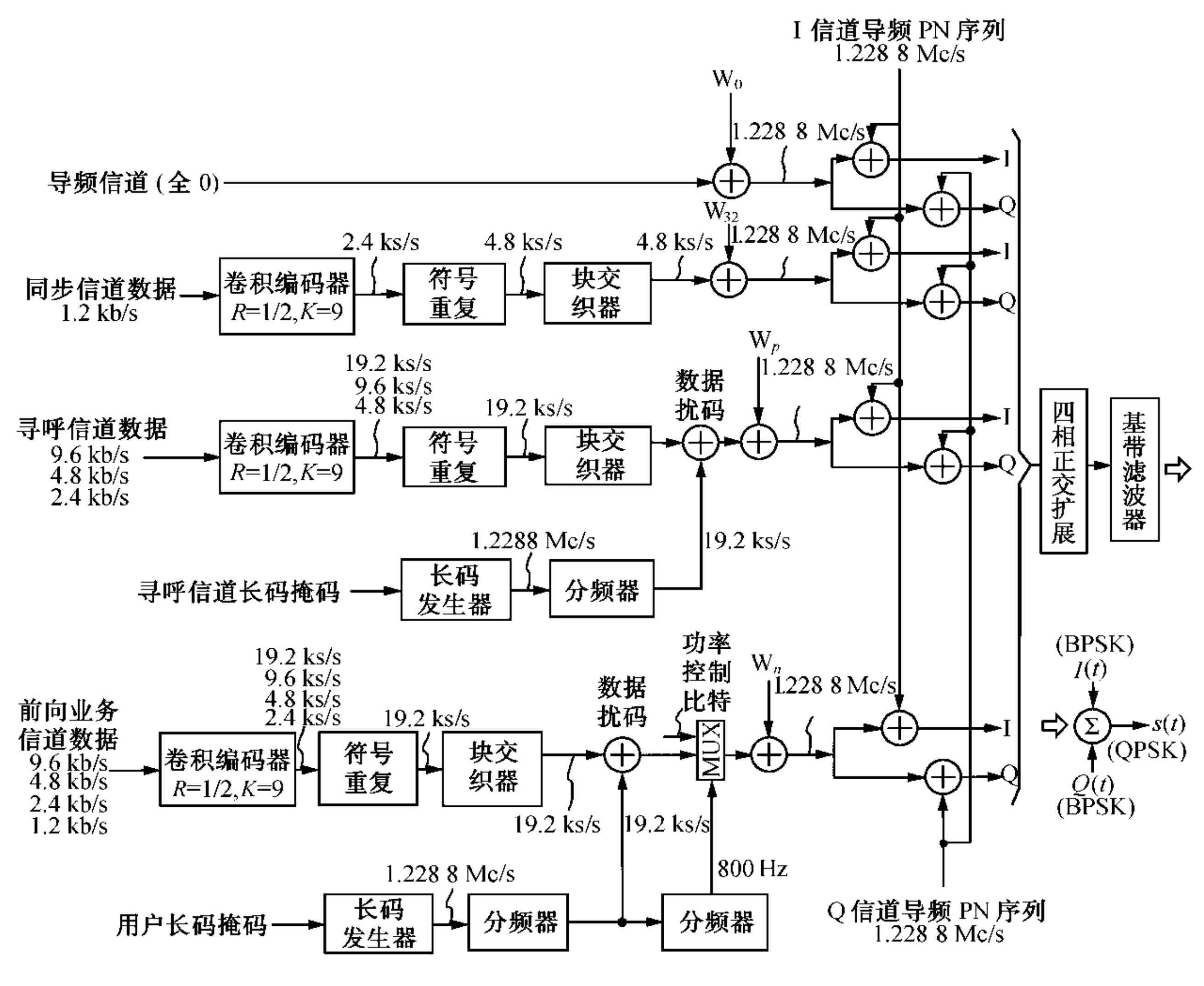


图 1.2　前向 CDMA 码道结构

编号为 32(W_{32})的码道分配给同步信道，用于向移动台传送同步消息。更重要的是，同步信道信号是经过编码、交织、扩频和调制的扩频信号，被移动台用来获取初始时间同步。

寻呼信道信号也是经过编码、交织、扩频和调制的扩频信号。寻呼信道供基站在呼叫建立阶段传输控制信息，编号从 1 到 7 的连续码道($W_1 \sim W_7$)分配给寻呼信道。

前向业务信道用来在通话过程中基站向特定移动台发送用户语音编码数据或其他业务数据和信令业务。在同一前向 CDMA 信道工作时，63 减去同步信道数和寻呼信道数就得到最大前向业务信道数。

信道输入的数据速率如下：

1. 导频信道以 19.2 kb/s 的速率发送全“0”。
2. 同步信道以固定速率 1 200 b/s 分帧传输。
3. 寻呼信道支持 9 600 b/s，4 800 b/s 或 2 400 b/s 三种不同速率。
4. 前向业务信道支持 9 600 b/s，4 800 b/s，2 400 b/s 或 1 200 b/s 四种变速率。

同步信道、寻呼信道和前向业务信道在发送之前都进行卷积编码以便纠错，而导频信道不需要卷积编码。编码符号通常定义为纠错编码器的输出。编码器输入信息比特，输出编码符号。每当信息速率低于 9 600 b/s 时，除了导频码道，其他所有码道的编码符号在进行块交织之前都重复若干次。

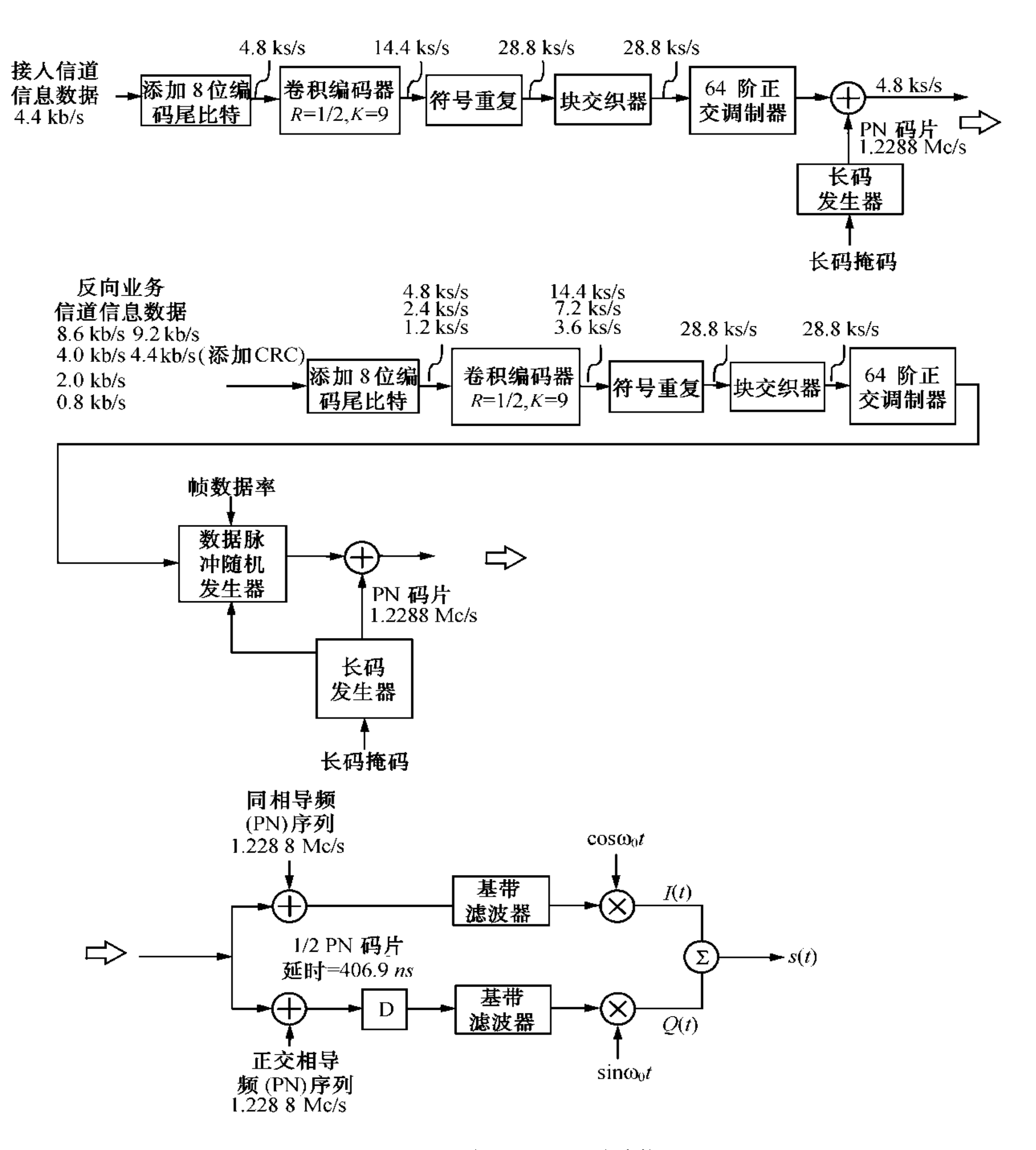


图 1.3 反向 CDMA 码道结构

对寻呼信道和前向业务信道而言,重复次数取决于每路信道的数据速率。速率为4.8 kb/s的编码符号要重复1次(每个符号连续出现2次)。速率为2.4 kb/s的编码符号要重复3次(每个符号连续出现4次)。速率为1.2 kb/s的编码符号要重复7次(每个符号连续出现8次)。这样,对于所有速率的数据(9 600 b/s,4 800 b/s,2 400 b/s或1 200 b/s),符号重复将使得调制符号速率统一为19.2 kb/s。对同步信道,编码符号重复1次(每个符号连续出现2次),这样调制符号速率为4 800 b/s。

同步信道、寻呼信道和前向业务信道重复后的所有符号进行块交织。使用块交织的目的是防止数据在发送到多径衰落环境中后发生突发性错误。交织之后,前向 CDMA 信道的每路码道用一个64阶 Walsh 函数进行正交扩频,再由导频 PN 序列正交对以1.228 8 Mc/s的速率扩频。正交扩

频的二进制数据(0 和 1)输入基带滤波器。基带滤波之后,前向 CDMA 信道将相应的 BPSK(二进相移键控)的调制数据和载波结合,产生 QPSK(正交相移键控)信号再发送。

数据扰码应用在寻呼信道和前向业务信道。数据扰码在块交织器输出端以 19.2 ks/s 传输。参见图 1.2,数据扰码通过对交织器输出数据与用户号码对应的长伪码进行模 2 加来完成。长码是周期为 $2^{42}-1$ 的 PN 序列,用于前向 CDMA 信道(即寻呼和前向业务信道)的扰码和反向 CDMA 信道(即接入和反向业务信道)的扩频。

长码发生器的掩码是长为 42 比特的二进制数,用以产生长码的惟一标识。长码的每个 PN 码片由长码发生器内的 42 比特 LFSR(左移寄存器)和一个 42 比特掩码的内积做模 2 运算而产生。长码工作在 1.228 8 MHz 的时钟速率,这和长码发生器的输出 PN 码片序列速率相等。请注意一个 PN 码片定义为 PN 序列的一个比特。如果长码以 64 比特(或码片)划分,那么第一个 64 比特将以 19.2 kb/s 速率用于数据扰码。图 1.2 所示的分频器功能为,以每输入 64($=1.2288\times10^6/192\times10^2$)比特输出 1 比特的速率减小长码的大小。

卷积码(n,k,m)由 k 位输入、n 位输出线性时序电路和 m 阶存储寄存器实现。

寻呼信道上基站不插入功率控制子信道。但是功率控制子信道信号在前向业务信道以 800 b/s的速度(即每 1.25 毫秒 1 比特)连续发送功率控制信息。功率控制比特 0 表示移动台必须增加平均输出功率控制级别,功率控制比特 1 表示移动台必须减少平均输出功率控制级别。这样,针对在前向业务信道收到的每个有效功率控制比特,移动台将调整其输出功率级别。

反向 CDMA 信道由接入信道和反向业务信道组成。图 1.3 显示了反向 CDMA 信道的完整结构。在反向 CDMA 信道发送的数据被封装成 20 毫秒帧。所有在反向 CDMA 信道上发送的数据都进行卷积编码(用于纠正随机错误)、块交织(用于防止突发错误)、含有 64 位长码片的 64 阶 Walsh 码调制,并在发送之前用周期为 $2^{42}-1$ 的长码直接序列扩展。

移动台在接入信道上发送数据时不使用数据脉冲随机发生器。但是在反向业务信道,使用数据突发随机发生器产生掩码比特 0 和 1,用于对码重复产生的冗余数据进行随机掩码。反向业务信道和接入信道由长码直接序列扩展。扩展操作包含对数据突发随机发生器的输出流和长码进行模 2 加。直接序列扩展之后,反向业务信道和接入信道进行正交相位扩展,如图 1.3 所示。请注意,由正交相位导频 PN 序列扩展的 Q 信道数据相对于同相导频 PN 序列扩展的 I 信道数据,存在半个码片(406.9 纳秒)的延时。

帧长定义为系统基本定时间隔。对于接入信道、寻呼信道和前向/反向业务信道,帧长为 20 毫秒。对同步信道,帧长为 26.666 毫秒。

帧质量指示器用于 9 600 b/s 和 4 800 b/s 速率的业务信道的 CRC 校验。CRC(循环冗余码)是一类线性错误检测码,能查找多项式余数而产生奇偶校验比特。帧质量指示器(CRC)支持接收机的两种功能。第一种是确定帧的发送速率,第二种是确定帧是否出错。

编码器尾比特表示加在数据帧末尾的固定比特序列,用于将卷积码编码器复位成已知状态。

无论是反向 CDMA 信道还是前向 CDMA 信道,发送的数据都被封装成 20 毫秒长的帧。

前向业务信道和反向业务信道帧结构都可以描述如下:

(Ⅰ)每路以下列速率发送的反向业务信道帧:

1. 9 600 b/s,含有 192 比特,由 172 位信息比特加上 12 位帧质量指示器(CRC)和 8 位编码器尾比特组成。
2. 4 800 b/s,含有 96 比特,由 80 位信息比特加上 8 位 CRC 比特和 8 位编码器尾比特组成。

3. 2 400 b/s，含有 48 比特，由 40 位信息比特加 8 位编码器尾比特组成。速率为2 400 b/s的反向业务信道帧不使用 8 位 CRC 比特。
4. 1 200 b/s，含有 24 比特，由 16 位信息比特加 8 位编码器尾比特组成。速率为1 200 b/s的反向业务信道帧不使用 8 位 CRC 比特。
5. 每路接入信道帧含有 96 比特（速率为 4 800 b/s 的 20 毫秒帧），由 88 位信息比特加 8 位编码器尾比特组成。接入信道帧不使用 8 位 CRC 比特。

（Ⅱ）每路以下列速率发送的前向业务信道帧：

1. 9 600 b/s，含有 192 比特，由 172 位信息比特加上 12 位帧质量指示器（CRC）和 8 位编码器尾比特组成。
2. 4 800 b/s，含有 96 比特，由 80 位信息比特加上 8 位帧质量指示器（CRC）和 8 位编码器尾比特组成。
3. 2 400 b/s，含有 48 比特，由 40 位信息比特加 8 位编码器尾比特组成。速率为2 400 b/s的反向业务信道帧不使用 8 位 CRC 比特。
4. 1 200 b/s，含有 24 比特，由 16 位信息比特加 8 位编码器尾比特组成。速率为1 200 b/s的反向业务信道帧不使用 8 位 CRC 比特。

前向业务信道帧长 20 毫秒，选择发送数据必须基于每帧进行。尽管基于帧的数据率可能不同，但是通过码重复可以保持调制速率统一为 19.2 kb/s。同步信道比特速率为1 200 b/s。同步信道帧长 26.666 毫秒，寻呼信道帧长 20 毫秒。

1.3 CDMA 信道链路的特点和功能

由于 CDMA 使用前向和反向信道，因此蜂窝小区覆盖范围内的基站和移动台之间存在功能上的关系。表 1.1 总结并解释了基站和移动台的功能以及 CDMA 信道所有码道的特点。这张表简要介绍了系统捕获、定时、同步、交织、正交信道化、扩展技术、功率控制、呼叫处理、切换过程、鉴权和消息保密。其中基站和移动台之间传递四种系统开销消息，以满足 CDMA 蜂窝系统所需的功能。

表 1.1 CDMA 蜂窝系统

系统开销消息	
系统参数消息	包含寻呼信道、注册参数、帮助导频获取的参数等
接入参数消息	包含接入信道和控制参数。其中一些控制参数提供了对移动台的动态反馈来控制其发射速率，这样有助于稳定接入信道
相邻基站列表消息	包含加速向相邻基站切换的信息，包括导频 PN 的时间偏移以及基本相邻基站配置
CDMA 信道列表消息	列出 CDMA 频率分配，包含寻呼信道。这样允许移动台正确确定何处找到寻呼信道

（续表）

移动台	信道	基站
	导频信道	
● 移动台一直监视导频信道，除非不在时隙化模式下接收。时隙化模式是指当移动台处于空闲状态时，移动台仅监视寻呼信道的选定时隙	● 导频信道是移动台用来捕获、定时的参考信道并提供相关解调的相位参考	● 基站连续为基站支持的每一路CDMA信道发送导频信道
● 移动台利用导频信道进行同步	● 基站在每一路激活的前向CDMA信道上始终发送导频信号。导频信道是不调制的扩频信号，用于基站覆盖范围内的移动台进行同步操作	● 导频和同步信道处理期间，基站发送导频和同步信道信息，移动台使用该信息捕获并同步到CDMA系统
● 移动台捕获CDMA系统导频信道并提供相关解调的相位参考		
	同步信道	
● 接收同步信道消息	● 同步信道用于系统捕获	● 发送同步信道消息
● 将定时调整到正常系统状态	● 一旦移动台捕获到系统，它一般不会重新使用同步信道	● 同步信道主要传输同步信道信息
● 确定和开始监视寻呼信道	● 由于每个基站的导频PN序列偏移都不同，每个基站的同步信道帧结构也不一样	● 同步信道帧等于导频PN序列长度
● 同步信道是经过编码、交织、扩展和调制的扩频信号，移动台用同步信道来捕获初始时间同步	● 同步信道向移动台传输同步消息	● 使用导频PN序列发送时间对齐的同步信道帧
	寻呼信道	
● 只监视单路寻呼信道	● 寻呼信道通过散列所有可用的寻呼信道确定	● 基站也可对特定寻呼信道指定移动台
● 指令（广义消息）用于控制特定移动台	● 数据率：2.4 kb/s，4.8 kb/s，9.6 kb/s（不同CDMA频率分配上的寻呼信道）	● 寻呼消息包含到一个或多个移动台的寻呼
● 指令用于从注册确认到锁定或阻止非法移动台发送的所有消息	● CDMA信道列表消息中列出的CDMA频率分配允许移动台正确确定何处找到其寻呼信道	● 当基站收到来自移动台的呼叫时发送寻呼，寻呼通常由不同基站发送
● 信道分配消息允许基站给移动台分配业务信道、改变其寻呼信道分配或引导移动台使用模拟调频系统	● 寻呼信道消息传送从基站到移动台的消息	● 寻呼信道是经过编码、交织、扩展和调制的扩频信号，用于从基站到移动台发送控制信息和寻呼
● 数据扰码用在寻呼信道，通过对交织器输出数据和长码进行模2加运算实现	● 主要消息类型：系统开销、寻呼、指令和信道分配	
	接入信道	
● 在相关的一路接入信道发送响应寻呼信道消息	● 接入信道为随机接入CDMA信道	● 通过一路相关的寻呼信道上的消息，在特定接入信道上响应发送
● 从一组可用接入信道集和PN时间序列中分别随机选择一路接入信道和一个PN时间序列	● 当移动台不使用业务信道时，提供从移动台到基站的通信	● 基站能接收多个同时发送，除非两个或者更多移动台选择相同接入信道和相同的PN时间序列

（续表）

● 当移动台不使用业务信道时，提供到基站的通信	● 在本书中仅使用4 800 b/s模式	● 控制接入信道发送速率，防止多个移动台同时进行发送
● 靠不同的长PN码区分	● 提供：呼叫来源、寻呼响应、指令和注册	● 接入信道发送的控制通过在寻呼信道上发送的接入参数消息完成
	● 每个寻呼信道和一个或多个接入信道配对	
	● 接入信道发送速率可以针对不同类型的发送和不同类型的移动台改变，但是必须选定一种特定数据率	
	业务信道	
● 当移动台被指定给某一业务信道时，直接在业务信道上发送信令（模糊的脉冲）	● 前向和反向业务信道使用类似的20毫秒帧控制结构。帧发送速率可以是9.6 kb/s，4.8 kb/s，2.4 kb/s或1.2 kb/s	● 业务信道处理中，基站使用前向和反向业务信道和移动台进行通信，同时移动台处于由移动台控制的业务信道状态之下
● 空白脉冲信令以9.6 kb/s发送，并将一个或多个主业务数据帧（通常为合成话音）替换为信令数据，就像模拟调频系统一样	● 业务信道存在5种类型的控制消息： 控制呼叫本身的消息 控制切换的消息 控制前向线路功率的消息 安全和鉴权的消息 发出来自或发往移动台信息的消息	● 业务信道处理包含下列一些子状态： 业务信道初始化子状态。该子状态下，基站开始在前向业务信道上发送数据并在反向业务信道上接收数据。 等待指令子状态。该子状态下，基站向移动台发送告警消息。 会话子状态。该子状态下，基站和移动台主服务应选用交换主业务比特。 释放子状态。该子状态下，基站断开呼叫连接
● 脉冲信令以9.6 kb/s的速率在一个帧内发送信令和主业务数据。在保证话音质量的前提下，脉冲信令的性能强于空闲脉冲信令	● 前向业务信道用于呼叫过程中从基站到特定移动台发送用户数据和信令业务	● 业务信道操作过程中，基站和移动台可以支持主业务服务。每一种可选服务具有一组要求，用来控制基站和移动台处理前向和反向业务信道帧中主业务比特的方式。 基站或移动台都可以要求可选服务。当寻呼移动台或业务信道操作时，基站可以要求特殊服务
● 移动台支持下列切换过程： 软切换：开始和新的基站进行通信，同时又不中断和原来基站的通信； 硬切换：在不同基站、不同载频、或不同帧偏移之间的通信	● 反向业务信道用于传输从信令移动台到单个或多个基站的用户数据或其他业务数据	
● 加密和鉴权： 只有当基站处理的共享私有数据集和移动台的相同时，鉴权才成功进行。128比特长的SSD存储在移动台。SSD分为两个不同子集：SSD-A和SSD-B。前64比特的SSD-A用于支持鉴权过程；后64比特的SSD-B用于支持CDMA话音保密和数据加密	● 前向业务信道应用数据扰码技术。反向业务信道应用数据扩展技术	

1.4　呼叫处理

呼叫处理可以分为两部分，移动台呼叫处理和基站呼叫处理。呼叫处理是指移动台和基站之间的消息流协议技术。

1.4.1　移动台呼叫处理

如图 1.4 所示，移动台呼叫处理包含以下四个状态。

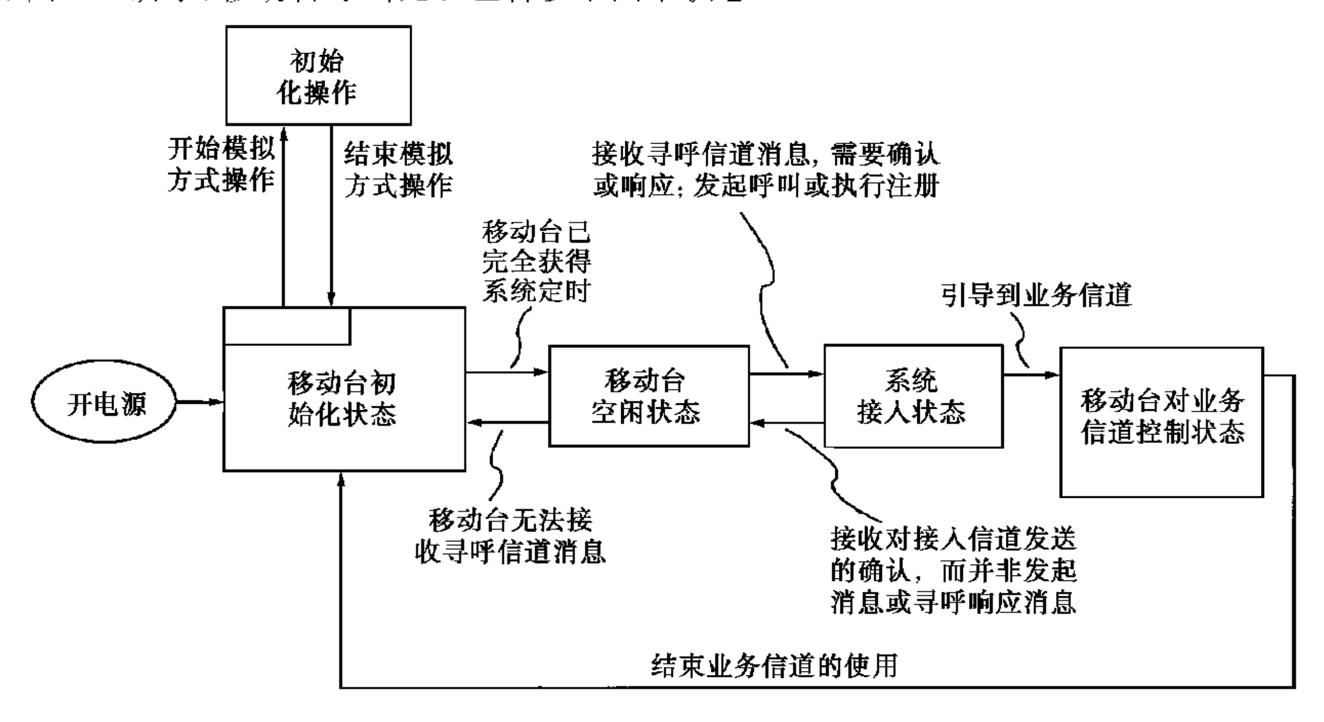


图 1.4　移动台呼叫处理状态

移动台初始化状态

在该状态下，移动台必须：

- 选择使用哪个系统（模拟或 CDMA 运行模式）
- 在 20 毫秒之内捕获所选 CDMA 系统的导频信道
- 接收并处理同步信道消息以获得系统配置和定时信息
- 将移动台长码定时和系统定时与 CDMA 的长码定时和系统定时保持同步

移动台空闲状态

该状态下，移动台监视寻呼信道消息。移动台可以接收消息、入呼叫、发起源呼叫、发起注册或发起消息传送。

系统接入状态

该状态下，移动台在接入信道向基站发送消息并在寻呼信道上接收来自基站的消息。系统接入状态包含下列子状态，且移动台必须：

- 监视寻呼信道直到它已经收到一组当前的配置消息
- 向基站发送呼叫源消息

- 向基站发送寻呼响应消息
- 发送对来自基站消息的响应
- 向基站发送注册消息
- 向基站发送数据脉冲消息

移动台使用随机接入过程在接入信道发送消息。发送某条消息和接收该消息的确认消息的完整过程称为一次接入尝试。每次接入尝试的发送称为接入探测。移动台在接入尝试的每次接入探测发送相同消息。每次接入探测包含接入报头(前同步)和接入信道消息封装。

移动台控制业务信道状态

该状态下,移动台使用前向和反向业务信道和基站通信。该状态包含下列子状态,且移动台必须:

- 检验移动台能接收前向业务信道并在反向业务信道开始发送
- 等待信息消息的指令
- 等待用户响应呼叫
- 在主服务可选应用下,与基站交换主业务包
- 在释放子状态下断开呼叫

1.4.2 基站呼叫处理

基站呼叫处理是指基站和移动台之间与消息流相关的方法。

基站呼叫处理包含下列类型的处理:

导频和同步信道处理

在导频和同步信道处理过程中,基站发送导频信道和同步信道而移动台处于初始化状态,移动台使用同步信道捕获和同步 CDMA 系统。

寻呼信道处理

在寻呼信道处理中,基站发送寻呼信道接收消息而移动台处于空闲和系统接入状态,移动台监视寻呼信道。

接入信道处理

在接入信道处理中,基站监视接入信道接收移动台发送的消息,而移动台处于系统接入状态。

业务信道处理

在业务信道处理中,基站使用前向和反向业务信道和移动台通信,而移动台处于移动台控制业务信道状态。

1.5 鉴权和消息加密

鉴权是指基站确认移动台身份的过程。只有当基站处理的共享私有数据集和移动台的相同

时，鉴权才成功进行。SSD 是存储在移动台的 128 比特长的数据。SSD 分为两个不同子集：SSD-A 和 SSD-B。64 比特的 SSD-A 用于支持鉴权过程；64 比特的 SSD-B 用于 CDMA 话音保密和数据加密。

基站配备数据库，该数据库含有针对每个蜂窝系统内注册移动台的鉴权密钥（A 密钥）和共享数据（SSD）。该数据库用于移动台鉴权。如果基站支持移动台鉴权，那么基站将发送和接收鉴权消息并执行鉴权计算。SSD 使用 SSD 产生过程进行更新，该过程由移动台特定信息（ESN）、随机数据（RANDSSD）和移动台 A 密钥进行初始化。64 比特 A 密钥存储在移动台永久安全和鉴权存储器中，并且只对本移动台和移动台相关的本地位置注册/鉴权中心（HLR/AC）公开。

SSD 更新过程如图 1.5 所示，执行方法如下所示。

基站在寻呼信道或前向业务信道上发送 SSD 更新消息指令。收到 SSD 更新消息指令之后，移动台将输入参数（RANDSSD、ESN、A 密钥）设置到 SSD 产生算法中。移动台然后执行 SSD 产生过程。SSD 产生过程的输出就是 SSD-A-NEW 和 SSD-B-NEW。然后移动台选择一个 32 比特随机数，并放在接入信道或反向业务信道的基站询问指令中向基站发送。

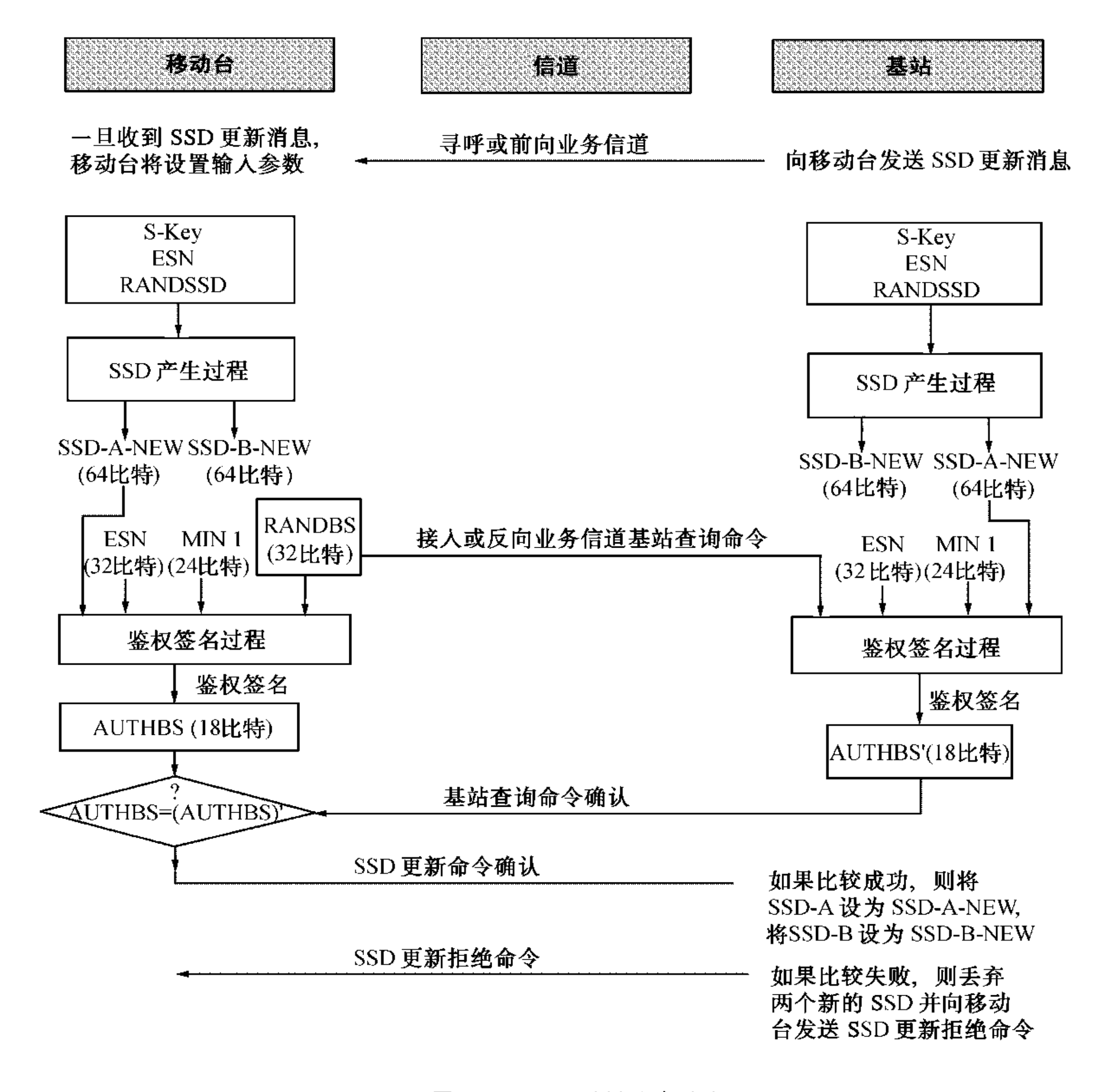


图 1.5　SSD 更新消息流程

移动台和基站都设置鉴权签名过程的输入参数(RANDBS,ESN,MIN1,SSD-A-NEW),如图 1.5 所示,并执行鉴权签名过程。

AUTHBS 设置成 18 比特的鉴权签名。基站将 AUTHBS 计算值放在寻呼信道或前向业务信道的基站询问确认指令中向移动台发送。收到基站询问确认指令后,移动台将收到的 AUTHBS 值和它内部产生的值进行比较。

如果比较成功,则移动台执行 SSD 更新过程,将 SSD-A 和 SSD-B 分别设置成 SSD-A-NEW 和 SSD-B-NEW。移动台然后向基站发送 SSD 更新过程确认指令,表明 SSD 更新过程成功完成。

如果比较失败,那么移动台丢弃 SSD-A-NEW 和 SSD-B-NEW。移动台之后向基站发送 SSD 更新过程拒绝指令,表明 SSD 更新过程不成功。

在收到 SSD 更新过程确认指令后,基站将 SSD-A 和 SSD-B 设置成 HLR/AC 计算的值。

SSD 更新仅在移动台和其相关的 HLR/AC 上进行,而不在服务系统中进行。服务系统保留 HLR/AC 通过和移动台 HLR/AC 系统间通信而计算的 SSD 值。

为了保护用户敏感信息,设计了一种方法对选定业务信道信令消息的特定字段进行加密。如果基站支持移动台鉴权,那么它可能也支持消息加密,通过发送加密控制消息并执行加密和解密操作来实现。然而,TIA/EIA/IS-95 不支持加密算法,因为算法受到美国国际交易和武器规则(ITAR)控制,该规则中有算法如何初始化和应用的描述。

如果鉴权没有执行,就不应该对信令加密。信令消息加密是对每个呼叫单独控制的。

加密过程初始化和使用细节的协议将在第 8 章论述。

CDMA 系统中通过用于 PN 扩展的私有长码掩码能提供话音保密。话音保密只能用在业务信道。所有呼叫都是使用 PN 扩展的公用长码掩码发起的。移动台用户可以在呼叫设置过程使用呼叫源消息或寻呼响应消息,或者在业务信道操作过程使用长码转换请求指令来要求话音保密。

如果鉴权没有执行,就不会进行私有长码掩码的转换。要发起私有或公用长码掩码转换,基站或移动台应在业务信道上发送一条长码转换请求指令。

1.6 宽带 CDMA(W-CDMA)信道

1.2 节和 1.3 节讲述的是基于 TIA/EIA/IS-95 的 CDMA 链路。表 1.2 列出了反向和前向信道的调制参数。

表 1.2 反向和前向信道的调制参数

参数 \ 信道	反向业务信道	前向业务信道
数据速率(b/s)	9 600,4 800,2 400,1 200	9 600,4 800,2 400,1 200
PN 码片速率(Mc/s)	1.228 8	1.228 8
码符号速率(s/s)	28 800	19 200

另外,TAG-7 于 1995 年发布了"Proposed Wideband CDMA PCS Standard",即 JTC(AIR)/95-01-30。无线接入联合技术委员会(JTC)成立的目的是评估八种不同个人通信系统(PCS),这些系统都基于 IS-95 系统,可以选择附加服务的扩展系统。表 1.3 列出了各自业务信道的反向和前向信息信道的调制参数。

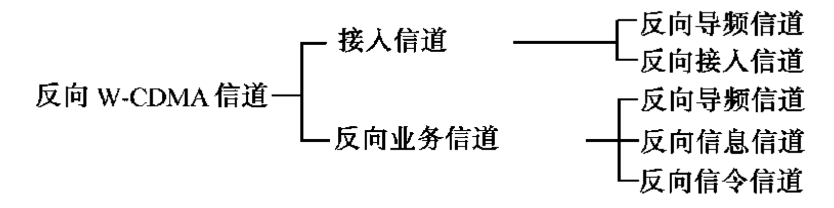
表 1.3 各自业务信道的反向和前向信息信道的调制参数

信道 / 参数	反向信息信道	前向信息信道
数据速率(b/s)	64 000,32 000,16 000	64 000,32 000,16 000
PN码片速率(Mc/s)	4.096	4.096
码符号速率(s/s)	64 000	64 000

1.6.1　反向 W-CDMA 信道

反向W-CDMA信道是从移动台到基站的通信链路。反向链路由接入和反向业务信道组成。每路信道包括一路导频信道。反向业务信道还包括一路信令信道。移动台在一路反向导频信道上发送导频信号,以便它和来自基站的导频信道信号同步。

接入信道包含两路信道,即反向导频信道和反向接入信道。反向业务信道包含三类信道,即反向导频信道、反向信息信道和反向信令信道,如下图所示。



这些信道使用CDMA直接序列技术共享相同的CDMA频率分配。每路信道靠不同的用户长码序列标识。

反向业务信道上的所有数据都经过卷积编码、交织和直接序列扩展调制再发送。图1.6显示了反向业务信道的反向CDMA信道结构。

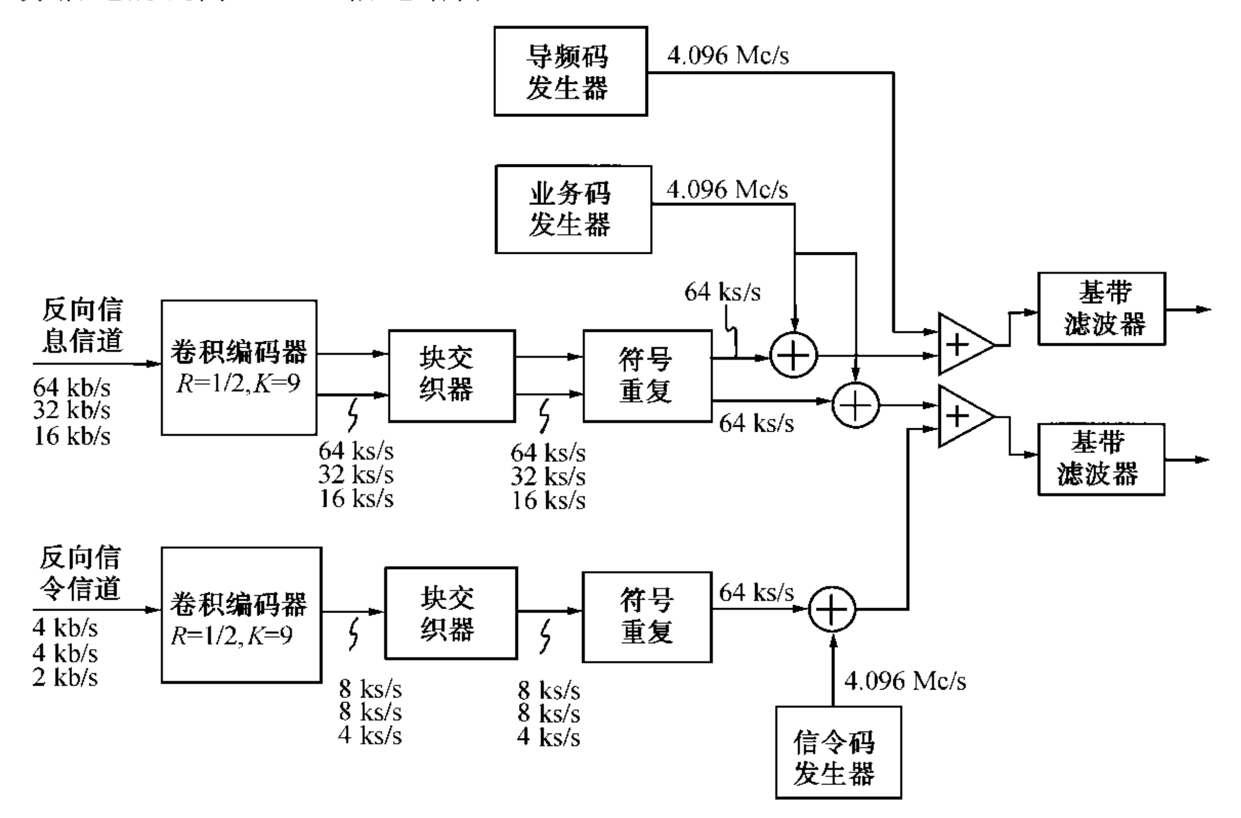
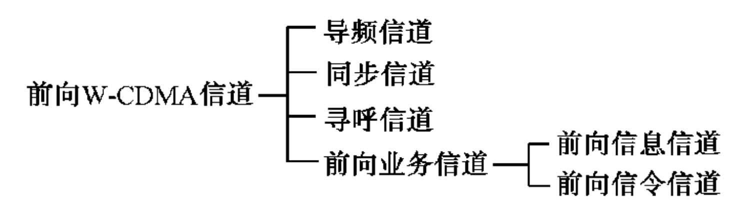


图1.6　反向W-CDMA反向业务信道结构

1.6.2　前向 W-CDMA 信道

前向W-CDMA信道包含下列码道:(1)一路导频信道,(2)一路同步信道,(3)最多八路寻呼信

道,(4)一定数量前向业务信道。前向业务信道包含一路前向信息信道和一路前向信令信道,如下图所示。



这些码道中每一路都由适当的 Walsh 码正交扩展,然后以 4.096 Mc/s 固定码片速率的 PN 序列扩展。图 1.7 显示了前向信息信道和前向信令信道的前向业务信道结构。

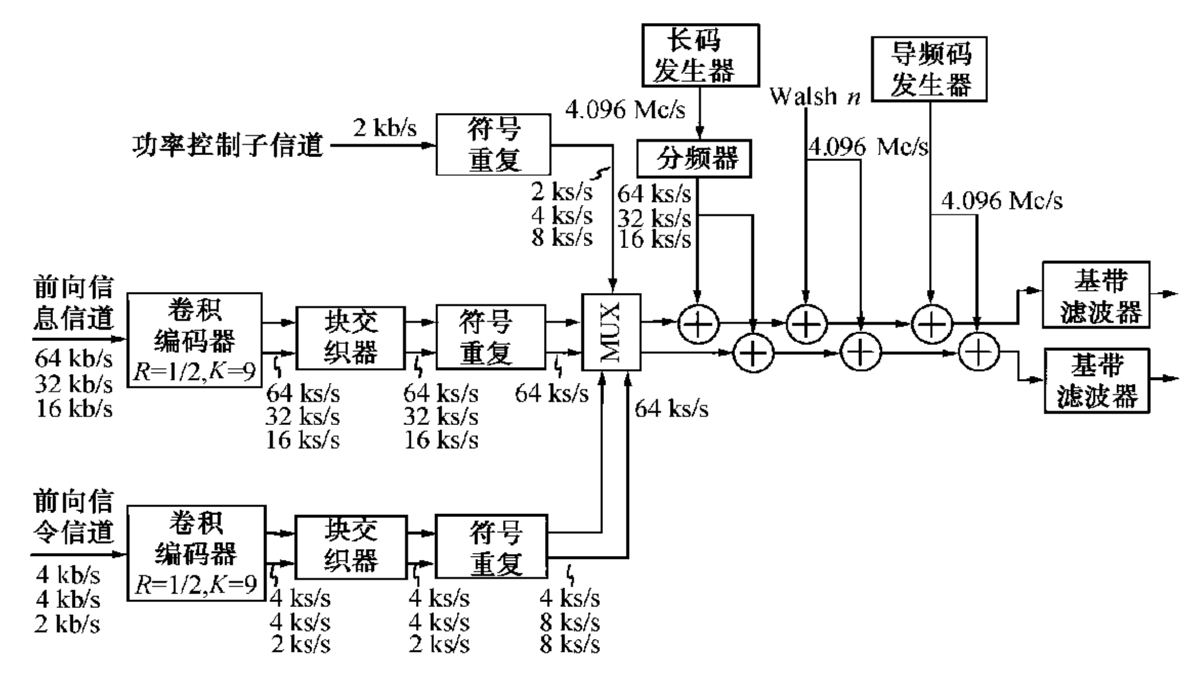


图 1.7 前向 W-CDMA 前向业务信道结构

- 导频信道激活的时候基站一直发送。导频信道发送的是不调制扩频信号,用于工作在该基站覆盖区域移动台的同步。导频信道必须用 Walsh 码 0 扩展再发送。
- 同步信道经过卷积编码、交织、扩展和调制成为扩频信号,用于工作在该基站覆盖区域移动台捕获初始帧同步。同步信道比特速率为 16 kb/s。同步信道的同相和正交相位导频 PN 序列使用相同的导频 PN 序列偏移作为给定站的导频信道。
- 寻呼信道经过编码、交织、扩展和调制,成为由工作在该基站覆盖区域移动台监视的扩频信号。基站利用多重寻呼信道发送系统信息和移动台指定消息。
- 前向业务信道用于在呼叫过程中向指定移动台发送用户数据和信令信息。给定前向信道能同时支持的最大前向业务信道数:数据率 64 kb/s 时为 64、数据率 32 kb/s 时为 128 和数据率 16 kb/s 时为 256。操作在相同前向信道的导频信道、同步信道和寻呼信道数必须从这些数中减去。
- 以数据率 64 kb/s,32 kb/s 和 16 kb/s 发送的前向业务信道帧的前向信息信道分别是 320 比特、160 比特和 80 比特。前向业务信道包含一路前向信息信道和一路前向信令信道。前向信令信道经过卷积编码、块交织、直接序列扩展、Walsh 函数正交扩展、以 4.096 Mc/s 固定码片速率的 PN 序列正交调制、基带滤波和 QPSK 波形发送。

第 9 章和第 10 章将详细讨论 W-CDMA 信道。

第 2 章 CDMA 信道操作技术基础

前向 CDMA 信道包含导频信道、同步信道、寻呼信道和前向业务信道。这些信道每路都经过适当的 Walsh 函数正交扩展，然后以 1.228 8 Mc/s 固定速率由正交相位导频 PN 序列扩展。

反向 CDMA 信道由接入信道和反向业务信道组成。反向信道上发送的所有数据都经过卷积编码、块交织、64 阶正交调制和长码直接序列扩展再发送。

本章介绍 CDMA 信道操作的几个最基本的组成部分。

2.1 卷积编码

现代数字通信系统常常设计成以非常高的速率传输。卷积码已应用于很多不同系统，例如，不仅在 CDMA 移动通信系统中应用卷积编码/译码，而且在空间和卫星也应用。为了防止系统出错，经常会使用卷积码。信息数据序列划分成许多长度为 k 的小块，每段小块被编码成长度为 n 的码字符号。卷积码(n,k,m)由 k 个输入、具有 m 阶存储的 n 个输出线性时序电路实现。通常，n 和 k 是较小的整数，且 $k<n$，但 m 相当大。特别地，当 $k=1$ 时，信息序列不再分成小块，以便可以连续处理。因此，卷积码的发展产生了很多有线和无线通信信道数字传输的实际应用。

卷积码(n, k, m)指定的码率为：$R=k/n$，编码器级数为 $m=K-1$，其中 K 是码的约束长度。编码器存储阶数等于数据序列时延。m 级 n 维生成序列集通常可以按如下方式描述：

$$g_i^{(j)}=(g_{i,0}^{(j)},g_{i,1}^{(j)},\cdots,g_{i,m}^{(j)}) \tag{2.1}$$

其中 $i=1, 2, \cdots, k$ 表示输入端数，$j=1, 2, \cdots, n$ 表示模 2 加法器数（输出端）。

方程（2.1）也可以以多项式形式表达为：

$$g_i^{(j)}(D)=\sum_{\lambda=0}^{m} g_{i,\lambda}^{(j)} D^{\lambda} \tag{2.2}$$

其中 D 是时延操作符，每一项 D 的幂对应于该项的单位时延数。

每个生成序列直接由从编码器级到各自模 2 加法器的连接序列确定，1 表示连接，0 表示断开。每个生成序列包含 $m+1$ 位二进制数。如果每次信息序列 $d^{(i)}=(d_0^{(i)},d_1^{(i)},d_2^{(i)},\cdots)$输入编码器 1 比特，那么编码器输出序列 $c^{(j)}=(c_0^{(j)},c_1^{(j)},c_2^{(j)},\cdots)$可以通过合并 $d^{(i)}$ 和 $g_i^{(j)}$ 离散卷积得到，即：

$$c^{(j)}=\sum_{i=1}^{k} d^{(i)} * g_i^{(j)}, j=1,2,\cdots,n \tag{2.3}$$

其中

$$c_{\lambda}^{(j)}=\sum_{l=1}^{m}\sum_{i=1}^{k} d_{\lambda-l}^{(i)} g_{i,l}^{(j)},\ l=0,1,\cdots,\lambda \tag{2.4}$$

基站对前向 CDMA 信道如同步、寻呼和业务信道上发送的数据进行卷积编码。前向 CDMA 信道使用码率 $R=1/2$、约束长度为 9 的$(2, 1, 8)$卷积码。

该卷积码的生成序列为：

$$g_1^{(1)}=753（八进制）=(111101011)（二进制）$$

$$g_1^{(2)}=561(\text{八进制})=(101110001)(\text{二进制})$$

由于码率为 1/2，所以每次编码器输入一位数据，编码器输出就产生两位编码符号。

初始化后第一个输出符号 $c_\lambda^{(1)}$ 是由生成序列 $g_1^{(1)}$ 编码的符号，第二个输出符号 $c_\lambda^{(2)}$ 是由生成序列 $g_1^{(2)}$ 编码的符号，如图 2.1 所示，卷积编码包含连续时延数据序列选定抽头的模 2 加。

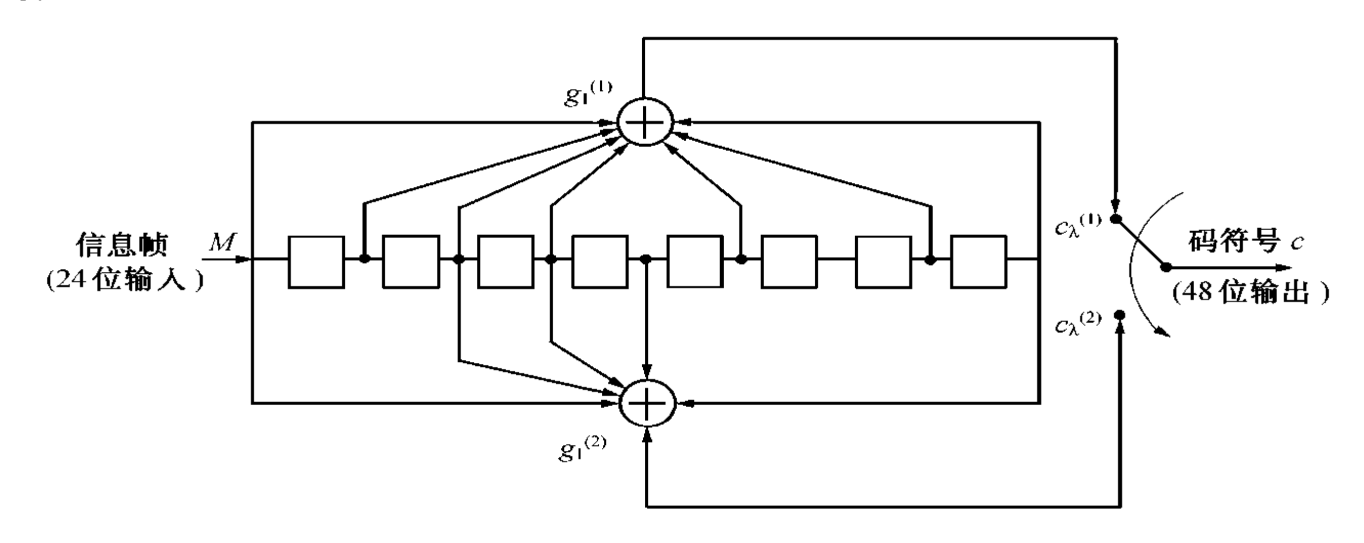


图 2.1　24 位输入的(2，1，8)卷积编码器

例 2.1　以 1.2 kb/s 速率传输的前向业务信道帧结构包含 24 比特(20 ms)。这 24 比特由 16 位信息比特和 8 位编码器尾比特组成，如图 2.2 所示。最后 8 位编码尾比特全部设成 0。如果信息序列表示成 $d=(1010100100000101)$，相应的多项式为 $d(x)=1+x^2+x^4+x^7+x^{13}+x^{15}$。由于 16 位信息比特后面有 8 位编码器比特，则前向业务信道帧表示为 $M=(101010010000010100000000)$ 或者写成多项式形式：

$$M(x)=1+x^2+x^4+x^7+x^{13}+x^{15}$$

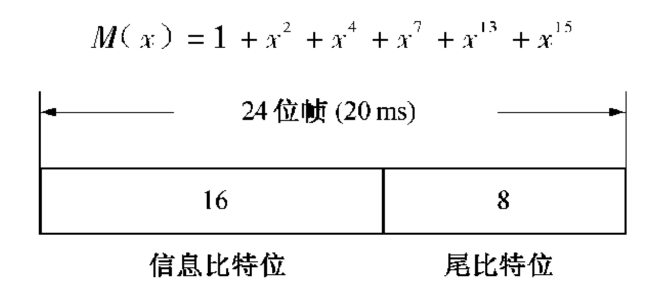


图 2.2　发送速率为 1 200 b/s 的前向业务信道帧

图 2.1 显示了用于该信道的(2，1，8)卷积码。由于 $m=8$，$n=2$，因此该编码器包含一个具有 2 个模 2 加法器的 8 级移位寄存器和用于编码器连续输出的转向器。对图 2.1 的卷积编码器，两个生成序列分别为：$g_1^{(1)}=(g_{1,0}^{(1)},g_{1,1}^{(1)},g_{1,2}^{(1)},g_{1,3}^{(1)},g_{1,4}^{(1)},g_{1,5}^{(1)},g_{1,6}^{(1)},g_{1,7}^{(1)},g_{1,8}^{(1)})=(111101011)$ 和 $g_1^{(2)}=(g_{1,0}^{(2)},g_{1,1}^{(2)},g_{1,2}^{(2)},g_{1,3}^{(2)},g_{1,4}^{(2)},g_{1,5}^{(2)},g_{1,6}^{(2)},g_{1,7}^{(2)},g_{1,8}^{(2)})=(101110001)$。

利用方程(2.4)，对于 $i=1$ 和 $j=1,2$，有：

$$c_\lambda^{(1)}=\sum_{l=0}^{8}d_{\lambda-l}^{(1)}g_{i,l}^{(1)}$$
$$c_\lambda^{(2)}=\sum_{l=0}^{8}d_{\lambda-l}^{(1)}g_{i,l}^{(2)}\qquad(2.5)$$

利用生成序列 $g_1^{(1)}$ 和 $g_1^{(2)}$ 的 $g_{1,l}^{(1)}$ 和 $g_{1,l}^{(2)}$，$0 \leqslant l \leqslant 8$，编码器每个模 2 加法器的输出分别为：

$$c_\lambda^{(1)} = d_\lambda^{(1)} + d_{\lambda-1}^{(1)} + d_{\lambda-2}^{(1)} + d_{\lambda-3}^{(1)} + d_{\lambda-5}^{(1)} + d_{\lambda-7}^{(1)} + d_{\lambda-8}^{(1)}$$

$$c_\lambda^{(2)} = d_\lambda^{(2)} + d_{\lambda-2}^{(2)} + d_{\lambda-3}^{(2)} + d_{\lambda-4}^{(2)} + d_{\lambda-8}^{(2)}$$

这样，输入帧 M 对应的编码符号计算如下：

λ	M	移位寄存器内容	$(c_\lambda^{(1)}, c_\lambda^{(2)}), 0 \leqslant \lambda \leqslant 23$
0	1	100000000	(11)
1	0	010000000	(10)
2	1	101000000	(00)
3	0	010100000	(01)
4	1	001010000	(10)
5	0	000101000	(01)
6	0	000010100	(01)
7	1	000001010	(00)
8	0	100000101	(00)
9	0	010000010	(00)
10	0	001000001	(00)
11	0	100100000	(00)
12	0	010010000	(11)
13	1	101001000	(10)
14	0	010100100	(01)
15	1	101010010	(11)
16	0	010101001	(00)
17	0	001010100	(10)
18	0	000101010	(11)
19	0	000010101	(10)
20	0	000001010	(00)
21	0	000000101	(11)
22	0	000000010	(10)
23	0	000000001	(11)

输出符号 $(c_\lambda^{(1)}, c_\lambda^{(2)})$ 连接成单个序列，即发送到块交织器的编码符号序列

$$\begin{aligned} c &= (c_0^{(1)} c_0^{(2)}, c_1^{(1)} c_1^{(2)}, \cdots, c_{23}^{(1)} c_{23}^{(2)}) \\ &= (111000011001010000000000111001110010111000111011) \end{aligned} \tag{2.6}$$

对于前向 CDMA 信道，每当信息速率小于 9 600 b/s 时，每个卷积编码的符号重复 k 次后再发送到块交织器。

信　　道	信息速率(b/s)	重复次数(k)	码重发次数
同步	1 200	1	2
寻呼	4 800	1	2
	9 600	0	1
前向业务	1 200	7	8

2 400	3	4
4 800	1	2
9 600	0	1

2.2 块交织

直接扩频 CDMA 支持同时在数量很大的用户群体而不光是单个用户之间的数字通信服务。这将反映在如何利用额外的维数和冗余来提高性能。使用了两种处理技术:用于提高额外冗余的交织技术和用于前向纠错的编码技术。

交织是排列符号序列的过程。这种为获得时间分集的重排过程称为交织,可以以两种方法考虑:块交织和卷积交织。交织也是一种为进行码字设计而纠正多径突发或长突发错误的有效技术。

在同步信道、寻呼信道和前向业务信道上,符号重复之后都进行块交织。同步信道使用块交织,输入符号序列由表 2.1 给出。表 2.1 有 16 行、8 列,因此调制符号速率为 4 800 s/s 时有 128 个调制符号。交织器时间跨度为 128/4800 = 26.66 ms。表 2.1 按行由上往下由左到右读。第一个输入符号 1 在最左上端,第二个输入符号 1 在第一个输入符号下面,第 17 个输入符号 9 在第一个输入符号右边,最后一个输入符号 64 位于最右下脚。另外,输出符号序列由表 2.2 给出。该表的读出顺序和表 2.1 一样。第一个输入符号 1 在最左上端,第二个输入符号 33 在第一个输入符号下面,第 17 个输入符号 3 在第一个输入符号右边,最后一个输入符号 64 位于最右下脚。重复编码符号按列写入交织器,直至完全填满矩阵。同步信道交织器是 16 行、8 列共 128 个元素的矩阵。因此,同步信道使用时间跨度为 26.66 ms 的块交织器,等价于以4 800 s/s速率调制符号的 128 个调制符号。

表 2.1 同步信道交织器输入(矩阵写操作)

1	9	17	25	33	41	49	57
1	9	17	25	33	41	49	57
2	10	18	26	34	42	50	58
2	10	18	26	34	42	50	58
3	11	19	27	35	43	51	59
3	11	19	27	35	43	51	59
4	12	20	28	36	44	52	60
4	12	20	28	36	44	52	60
5	13	21	29	37	45	53	61
5	13	21	29	37	45	53	61
6	14	22	30	38	46	54	62
6	14	22	30	38	46	54	62
7	15	23	31	39	47	55	63
7	15	23	31	39	47	55	63
8	16	24	32	40	48	56	64
8	16	24	32	40	48	56	64

类似地,寻呼信道和前向业务信道使用 24 行、16 列的块交织器,等价于以 19 200 s/s 速率调制符号的 384 个调制符号。可以看出交织器的时间跨度为 384/19200 = 20 ms。第 4 章将深入解释这点(4.12 节、表 4.9)。

例 2.2 考虑数据率为 1 200 b/s、含有 24 比特帧结构的前向业务信道,信道帧 M = (101010010000010100000000)输入到卷积码(2,1,8)编码器(与例 2.1 相同),该编码器输出

编码序列为 c =（1110000110010100000000000111001110010111000111011），该序列重复之后将作为块交织器的输入序列。

前向业务信道交织器输出速率为1 200 b/s，如表4.18所示。利用表4.18，按最左边两列从左到右往下读可以得到交织器输出 w。

表2.2　同步信道交织器输出（矩阵读操作）

1	3	2	4	1	3	2	4
33	34	34	36	33	35	34	36
17	19	18	20	17	19	18	20
49	51	50	52	49	51	50	52
9	11	10	12	9	11	10	12
41	43	42	44	41	43	42	44
25	27	26	28	25	27	26	28
57	59	58	60	57	59	58	60
5	7	6	8	5	7	6	8
37	39	38	40	37	39	38	40
21	23	22	24	21	23	22	24
53	55	54	56	53	55	54	56
13	15	14	16	13	15	14	16
45	47	46	48	45	47	46	48
29	31	30	32	29	31	30	32
61	63	62	64	61	63	62	64

1　9 17 25 33 41 5 13 21 29 37 45 3 11 19 27 35 43 7 15 23 31 39 47

2 10 18 26 34 42 6 14 22 30 38 46 4 12 20 28 36 44 8 16 24 32 40 48

w =（110100000011100111000111100100010110010001100101）

由于具有相同编号的符号表示重复码符号，交织器输出是 w 的7次相同重复序列，如表2.3所示。

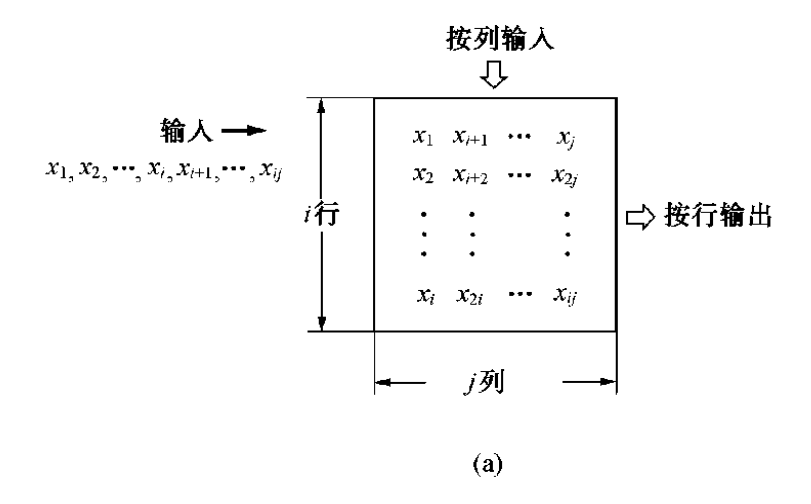
表2.3　前向业务信道1 200 b/s速率的交织器输出

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

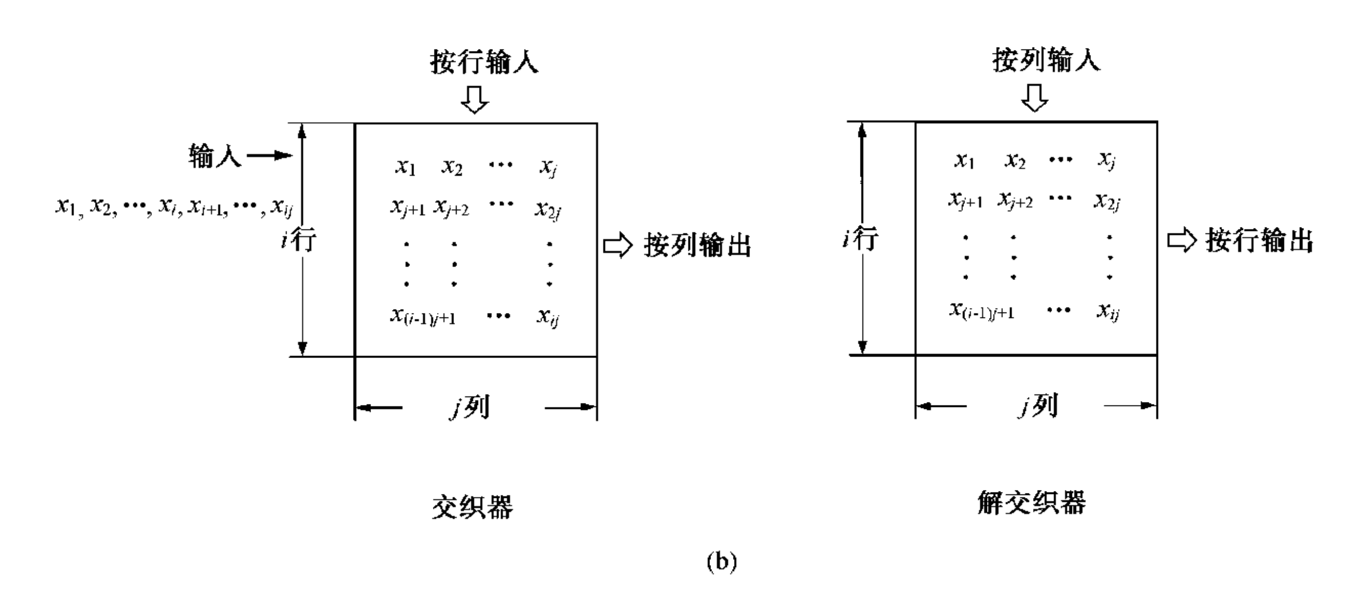
（续表）

1 0	1 0	1 0	1 0	1 0	1 0	1 0	1 0
1 0	1 0	1 0	1 0	1 0	1 0	1 0	1 0
0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1
0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0
1 0	1 0	1 0	1 0	1 0	1 0	1 0	1 0
1 0	1 0	1 0	1 0	1 0	1 0	1 0	1 0
1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1
0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1
0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0
0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0
1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1
1 0	1 0	1 0	1 0	1 0	1 0	1 0	1 0
1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1

块交织是数据顺序重排过程，可以如下考虑：交织后的输出符号（长度是卷积编码器编码符号的 i 倍）可以通过交织（或交错）获得，如表 2.1 和表 2.2 所示。实现方法是通过将 j 个编码符号安排成 i 行矩阵，然后按列（或行）发送符号，如图 2.3 所示。如果卷积码能纠正随机错误，那么交织器能纠正小于或等于 j 长度的突发错误。



(a)



(b)

图 2.3 块交织器的数据重排过程

2.3　Walsh 函数正交扩展

前向 CDMA 信道发送的每个码道都用一个 Walsh 函数以固定的 1.228 8 Mc/s 速率扩展，并在所有码道间提供正交信道化。反向 CDMA 信道使用 64 阶正交调制。调制符号是用 Walsh 函数产生的 64 个相互正交波形之一。每 6 个符号对应传输 64 种可能的调制符号之一，即 $c_i, 0 \leqslant i \leqslant 5$。

利用哈达码矩阵 $\boldsymbol{H}_n$，其中 n 是 2 的幂，即 $n = 2^m$，Walsh 函数按下列方式构造：一个哈达码矩阵是由 +1 和 −1 为元素组成的 $n \times n$ 正交矩阵，它的性质是任何不同两行有 $n/2$ 个位置对应值不同；矩阵中有一行含全 +1，其他行含有数目相等的 +1 和 −1，各占 $n/2$；而且，$\boldsymbol{H}_n$ 第一行和第一列元素全为 +1，只有当 n 是 2 的幂时才存在哈达码矩阵 $n \times n$。

将 +1 替换为 0，−1 替换为 1，$n = 2$ 时，哈达码矩阵表示为：

$$\boldsymbol{H}_2 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix} = \begin{bmatrix} 00 \\ 01 \end{bmatrix}$$

$n = 2^6$ 时的上述哈达码矩阵就是 64 × 64 正交 Walsh 函数，如表 2.4 所示。这个 64 乘 64 矩阵可以通过下列递归过程产生：

$$n = 2^m, 0 \leqslant m \text{（正整数）}$$

$$m = 0 \text{ 时}, \boldsymbol{H}_1 = 0$$

$$m = 1 \text{ 时}, \boldsymbol{H}_2 = \begin{bmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_1 \\ \boldsymbol{H}_1 & \overline{\boldsymbol{H}_1} \end{bmatrix} = \begin{bmatrix} 00 \\ 01 \end{bmatrix}$$

$$m = 2 \text{ 时}, \boldsymbol{H}_4 = \begin{bmatrix} \boldsymbol{H}_2 & \boldsymbol{H}_2 \\ \boldsymbol{H}_2 & \overline{\boldsymbol{H}_2} \end{bmatrix} = \begin{bmatrix} 0000 \\ 0101 \\ 0011 \\ 0110 \end{bmatrix}$$

$$m = 3 \text{ 时}, \boldsymbol{H}_8 = \begin{bmatrix} \boldsymbol{H}_4 & \boldsymbol{H}_4 \\ \boldsymbol{H}_4 & \overline{\boldsymbol{H}_4} \end{bmatrix} = \begin{bmatrix} 00000000 \\ 01010101 \\ 00110011 \\ 01100110 \\ 00001111 \\ 01101010 \\ 00111100 \\ 01101001 \end{bmatrix}$$

$$\vdots$$

$$\boldsymbol{H}_{2n} = \begin{bmatrix} \boldsymbol{H}_n & \boldsymbol{H}_n \\ \boldsymbol{H}_n & \overline{\boldsymbol{H}_n} \end{bmatrix}$$

其中 $2n = 2^{m+1}$，$\overline{\boldsymbol{H}_n}$ 为 $\boldsymbol{H}_n$ 的二进制互补矩阵。Walsh 矩阵 $\boldsymbol{H}_{64}$ 的行形成了分组长度为 $n = 64$ 的线性码，最小距离为 $d_{\min} = n/2 = 32$。

前向 CDMA 信道发送的每个码道由一个 Walsh 函数以固定码片速率 1.228 8 Mc/s 扩展在所有码

道中提供正交信道化。Walsh 函数扩展序列重复周期为 52.083(=64/1.2288 Mc/s)μs，等于一个前向业务信道调制符号的持续时间。

因此，使用 Walsh 函数 n(哈达码-Walsh 矩阵行数 n)扩展的码道分配码道编号 $n(0\leqslant n\leqslant 63)$。码道编号 0 总是分配给导频信道。如果存在同步信道，则分配码道编号 32。如果存在寻呼信道，则分配连续 1 到 7 的编号。剩下的码道分配给前向业务信道。这样，这些码道每路都经过适当的 Walsh 函数正交扩展，然后以 1.2288 Mc/s 的固定速率由正交导频 PN 序列扩展。

表 2.4　64阶正交符号集

符号内 Walsh 码片

调制符号索引			11	1111	1111	2222	2222	2233	3333	3333	4444	4444	4455	5555	5555	6666
	0123	4567	8901	2345	6789	0123	4567	8901	2345	6789	0123	4567	8901	2345	6789	0123
0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
1	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101
2	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011
3	0110	0110	0110	0110	0110	0110	0110	0110	0110	0110	0110	0110	0110	0110	0110	0110
4	0000	1111	0000	1111	0000	1111	0000	1111	0000	1111	0000	1111	0000	1111	0000	1111
5	0101	1010	0101	1010	0101	1010	0101	1010	0101	1010	0101	1010	0101	1010	0101	1010
6	0011	1100	0011	1100	0011	1100	0011	1100	0011	1100	0011	1100	0011	1100	0011	1100
7	0110	1001	0110	1001	0110	1001	0110	1001	0110	1001	0110	1001	0110	1001	0110	1001
8	0000	0000	1111	1111	0000	0000	1111	1111	0000	0000	1111	1111	0000	0000	1111	1111
9	0101	0101	1010	1010	0101	0101	1010	1010	0101	0101	1010	1010	0101	0101	1010	1010
10	0011	0011	1100	1100	0011	0011	1100	1100	0011	0011	1100	1100	0011	0011	1100	1100
11	0110	0110	1001	1001	0110	0110	1001	1001	0110	0110	1001	1001	0110	0110	1001	1001
12	0000	1111	1111	0000	0000	1111	1111	0000	0000	1111	1111	0000	0000	1111	1111	0000
13	0101	1010	1010	0101	0101	1010	1010	0101	0101	1010	1010	0101	0101	1010	1010	0101
14	0011	1100	1100	0011	0011	1100	1100	0011	0011	1100	1100	0011	0011	1100	1100	0011
15	0110	1001	1001	0110	0110	1001	1001	0110	0110	1001	1001	0110	0110	1001	1001	0110
16	0000	0000	0000	0000	1111	1111	1111	1111	0000	0000	0000	0000	1111	1111	1111	1111
17	0101	0101	0101	0101	1010	1010	1010	1010	0101	0101	0101	0101	1010	1010	1010	1010
18	0011	0011	0011	0011	1100	1100	1100	1100	0011	0011	0011	0011	1100	1100	1100	1100
19	0110	0110	0110	0110	1001	1001	1001	1001	0110	0110	0110	0110	1001	1001	1001	1001
20	0000	1111	0000	1111	1111	0000	1111	0000	0000	1111	0000	1111	1111	0000	1111	0000
21	0101	1010	0101	1010	1010	0101	1010	0101	0101	1010	0101	1010	1010	0101	1010	0101
22	0011	1100	0011	1100	1100	0011	1100	0011	0011	1100	0011	1100	1100	0011	1100	0011
23	0110	1001	0110	1001	1001	0110	1001	0110	0110	1001	0110	1001	1001	0110	1001	0110
24	0000	0000	1111	1111	1111	1111	0000	0000	0000	0000	1111	1111	1111	1111	0000	0000
25	0101	0101	1010	1010	1010	1010	0101	0101	0101	0101	1010	1010	1010	1010	0101	0101
26	0011	0011	1100	1100	1100	1100	0011	0011	0011	0011	1100	1100	1100	1100	0011	0011
27	0110	0110	1001	1001	1001	1001	0110	0110	0110	0110	1001	1001	1001	1001	0110	0110
28	0000	1111	1111	0000	1111	0000	0000	1111	0000	1111	1111	0000	1111	0000	0000	1111
29	0101	1010	1010	0101	1010	0101	0101	1010	0101	1010	1010	0101	1010	0101	0101	1010
30	0011	1100	1100	0011	1100	0011	0011	1100	0011	1100	1100	0011	1100	0011	0011	1100
31	0110	1001	1001	0110	1001	0110	0110	1001	0110	1001	1001	0110	1001	0110	0110	1001
32	0000	0000	0000	0000	0000	0000	0000	0000	1111	1111	1111	1111	1111	1111	1111	1111
33	0101	0101	0101	0101	0101	0101	0101	0101	1010	1010	1010	1010	1010	1010	1010	1010
34	0011	0011	0011	0011	0011	0011	0011	0011	1100	1100	1100	1100	1100	1100	1100	1100
35	0011	0110	0110	0110	0110	0110	0110	0110	1001	1001	1001	1001	1001	1001	1001	1001

（续表）

调制符号索引			11	1111	1111	2222	2222	2233	3333	3333	4444	4444	4455	5555	5555	6666
	0123	4567	8901	2345	6789	0123	4567	8901	2345	6789	0123	4567	8901	2345	6789	0123
36	0000	1111	0000	1111	0000	1111	0000	1111	1111	0000	1111	0000	1111	0000	1111	0000
37	0101	1010	0101	1010	0101	1010	0101	1010	1010	0101	1010	0101	1010	0101	1010	0101
38	0011	1100	0011	1100	0011	1100	0011	1100	1100	0011	1100	0011	1100	0011	1100	0011
39	0110	1001	0110	1001	0110	1001	0110	1001	1001	0110	1001	0110	1001	0110	1001	0110
40	0000	0000	1111	1111	0000	0000	1111	1111	1111	1111	0000	0000	1111	1111	0000	0000
41	0101	0101	1010	1010	0101	0101	1010	1010	1010	1010	0101	0101	1010	1010	0101	0101
42	0011	0011	1100	1100	0011	0011	1100	1100	1100	1100	0011	0011	1100	1100	0011	0011
43	0110	0110	1001	1001	0110	0110	1001	1001	1001	1001	0110	0110	1001	1001	0110	0110
44	0000	1111	1111	0000	0000	1111	1111	0000	1111	0000	0000	1111	1111	0000	0000	1111
45	0101	1010	1010	0101	0101	1010	1010	0101	1010	0101	0101	1010	1010	0101	0101	1010
46	0011	1100	1100	0011	0011	1100	1100	0011	1100	0011	0011	1100	1100	0011	0011	1100
47	0110	1001	1001	0110	0110	1001	1001	0110	1001	0110	0110	1001	1001	0110	0110	1001
48	0000	0000	0000	0000	1111	1111	1111	1111	1111	1111	1111	1111	0000	0000	0000	0000
49	0101	0101	0101	0101	1010	1010	1010	1010	1010	1010	1010	1010	0101	0101	0101	0101
50	0011	0011	0011	0011	1100	1100	1100	1100	1100	1100	1100	1100	0011	0011	0011	0011
51	0110	0110	0110	0110	1001	1001	1001	1001	1001	1001	1001	1001	0110	0110	0110	0110
52	0000	1111	0000	1111	1111	0000	1111	0000	1111	0000	1111	0000	0000	1111	0000	1111
53	0101	1010	0101	1010	1010	0101	1010	0101	1010	0101	1010	0101	0101	1010	0101	1010
54	0011	1100	0011	1100	1100	0011	1100	0011	1100	0011	1100	0011	0011	1100	0011	1100
55	0110	1001	0110	1001	1001	0110	1001	0110	1001	0110	1001	0110	0110	1001	0110	1001
57	0000	0000	1111	1111	1111	1111	0000	0000	1111	1111	0000	0000	0000	0000	1111	1111
57	0101	0101	1010	1010	1010	1010	0101	0101	1010	1010	0101	0101	0101	0101	1010	1010
58	0011	0011	1100	1100	1100	1100	0011	0011	1100	1100	0011	0011	0011	0011	1100	1100
59	0110	0110	1001	1001	1001	1001	0110	0110	1001	1001	0110	0110	0110	0110	1001	1001
60	0000	1111	1111	0000	1111	0000	0000	1111	1111	0000	0000	1111	0000	1111	1111	0000
61	0101	1010	1010	0101	1010	0101	0101	1010	1010	0101	0101	1010	0101	1010	1010	0101
62	0011	1100	1100	0011	1100	0011	0011	1100	1100	0011	0011	1100	0011	1100	1100	0011
63	0110	1001	1001	0110	1001	0110	0110	1001	1001	0110	0110	1001	0110	1001	1001	0110

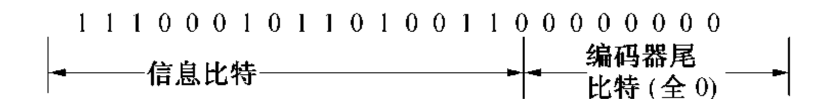
对于反向 CDMA 信道，调制符号将是由 Walsh 函数产生的 64 个相互正交波形之一。由于交织器输出编码符号速率为 28.8 kb/s，正交调制器输出符号速率变为 4.8 ks/s 或307.02 kc/s。这些编号为 0 到 63 的符号根据下列公式所得的调制符号指数进行选择：

$$\mathrm{MSI} = c_0 + 2c_1 + 4c_2 + 8c_3 + 16c_4 + 32c_5$$

其中 MSI 是调制符号指数；c_i，$0 \leqslant i \leqslant 5$，代表每组来自块交织器输出的 6 个二进制编码符号。每 6 个编码符号和一个调制符号相关联。传输单个调制符号需要的时间间隔等于1/4800 = 208.333 μs。相关调制符号的 1/64 时间称为 Walsh 码片，等于 208.333/64 = 3.2552 μs。一个 Walsh 码内，Walsh 码片以 0，1，2，…，63 的顺序发送。

例 2.3　速率为 1.2 kb/s 的反向 CDMA 信道帧由16 位信息比特和 8 位尾比特组成。移动台由接入信道或反向业务信道发送 24 比特帧数据卷积编码。卷积码（3，1，8）编码器生成序列为：

$g_1^{(1)}$ =（111001001），$g_1^{(2)}$ =（110110011），$g_1^{(3)}$ =（111001001）。假设输入卷积编码器的 24 比特帧序列为：



由于码率为 $R = 1/3$，故每输入 1 位帧比特，编码器输出端将产生 3 位编码符号（c_0, c_1, c_2）。模 2 加法器产生的码符号分别为：

c_0 = （110000111000011100000001）

c_1 = （100000001110101010100101）

c_2 = （101011001111101000001011）

这样，串联起来后编码符号输出可以写成如下：

c = （1111000010 0000100110 0100111011 0110010111

0011110001 0000010000 0010100011 11）

这就是输入交织器的符号。交织长度为 20 ms 的块交织器将对 c 内所有编码符号交织，然后进行正交调制。

以 1■200 b/s 速率发送的交织器输出可以根据下列交织器行顺序计算：

行 编 号

1 9 2 10 3 11 4 12 5 13 6 14 7 15 8 16

17 25 18 26 19 27 20 28 21 29 22 30 23 31 24 32

例如，利用交织器输入 c，对应于行编号为 1 的交织器输出为 101000111001000011。利用交织器输入 c 和表 2.5 中的行编号，可得到如表 2.6 所示的交织器输出 w。

表 2.5 反向业务信道交织器存储内容（**1200 b/s**）（**TIA/EIA/IS-95**）

1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	69
1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	69
1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	69
1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	69
1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	69
1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	69
1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	69
1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	69
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70
3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63	67	71
3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63	67	71
3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63	67	71
3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63	67	71
3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63	67	71
3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63	67	71
3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63	67	71

（续表）

3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63	67	71
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72

考虑表 2.6 中头两行（第 1 行和第 9 行），第 1 行为 101000111001000011，第 2 行为 100011110101100001。

表 2.6 对应于行编号的交织器输出

行编号	交织器输出	行编号	交织器输出
[1]	101000111001000011	[17]	100110100110000101
[9]	100011110101100001	[25]	100000011110010001
[2]	101000111001000011	[18]	100110100110000101
[10]	100011110101100001	[26]	100000011110010001
[3]	101000111001000011	[19]	100110100110000101
[11]	100011110101100001	[27]	100000011110010001
[4]	101000111001000011	[20]	100110100110000101
[12]	100011110101100001	[28]	100000011110010001
[5]	101000111001000011	[21]	100110100110000101
[13]	100011110101100001	[29]	100000011110010001
[6]	101000111001000011	[22]	100110100110000101
[14]	100011110101100001	[30]	100000011110010001
[7]	101000111001000011	[23]	100110100110000101
[15]	100011110101100001	[31]	100000011110010001
[8]	101000111001000011	[24]	100110100110000101
[16]	100011110101100001	[32]	100000011110010001

下面来看看如何计算对应第 1 行和第 9 行的调制符号：

1. 计算第 1 行

- 对于 6 个符号输入 101000（$c_0 = 1, c_1 = 0, c_2 = 1, c_3 = 0, c_4 = 0, c_5 = 0$），可以得到 MSI = 5。利用表 2.4，可以得到下列调制符号：

  01011010010110100101101001011010

  01011010010110100101101001011010

- 对于 6 个符号 111001，有 MSI = 39。对应 MSI = 39 的调制符号为：

  01101001011010010110100101101001

  10010110100101101001011010010110

- 对最后 6 个符号，得到 MSI = 48，相应调制符号为：

  00000000000000001111111111111111

  11111111111111110000000000000000

2. 计算第 9 行

- 输入符号 = 100011，MSI = 49

  调制符号 = 01010101010101011010101010101010

  10101010101010100101010101010101

- 输入符号 = 110101，MSI = 43

  调制符号 = 01100110100110010110011010011001

  10011001011001101001100101100110

- 输入符号 = 100001，MSI = 33

  调制符号 = 01010101010101010101010101010101

  10101010101010101010101010101010

这样可以得到关于第 1 行和第 9 行的交织器输出流：

$$w = 101000111001000011100011110101100001$$

利用表 2.4，对应于第 1 行和第 9 行的调制符号计算为：

0101101001011010010110100101101001011010010110100101101001011010

0110100101101001011010010110100110010110100101101001011010010110

0000000000000000111111111111111111111111111111110000000000000000

0101010101010101101010101010101010101010101010100101010101010101

0110011010011001011001101001100110011001011001101001100101100110

0101010101010101010101010101010110101010101010101010101010101010

一共有 $3072\times8 = 24576$ 个符号产生作为 Walsh 函数输出序列。正如期望的一样，发送这 24 576 个符号的时间周期等于 $24576/1.2288\times10^6 = 20$ ms。

2.4 直接序列扩展

反向 CDMA 信道中，反向业务信道和接入信道将由长码直接序列扩展，以提供有限的保密性。对反向业务信道，直接序列（DS）扩展操作包含对数据脉冲随机发生器输出数据和长码模 2 加。数据脉冲随机发生器产生掩码符号 0 和 1，随机地掩蔽由于码重复产生的冗余数据。掩码符号由帧数据率和长码的最后 14 比特位确定。对于接入信道，DS 扩展操作包含 64 阶正交调制器输出和长码的模 2 加。

设 $d_w(t)$ 为 Walsh 码片调制的数据序列，T_b 为数据比特时间间隔。Walsh 调制的数据序列被长码 $c(t)$ 的扩展 PN 码片模 2 加。每个 $c(t)$ 脉冲称为码片，T_c 表示码片时间间隔，$T_b=4T_c$。扩展 PN 码片序列速率固定在 1.228■8 Mc/s。由于 6 个编码符号由 64 个时间正交 Walsh 函数之一调制，调制符号发送速率固定在 28.8/6 = 4.8 ks/s。因此，每个 Walsh 码片由 4 个 PN 码片扩展，即 $1.2288\times10^6/307.2\times10^3 = 4$。由 1.228■8 Mc/s 长码 PN 码片相乘的直接序列 $d_w(t)$ 如图 2.4 所示。

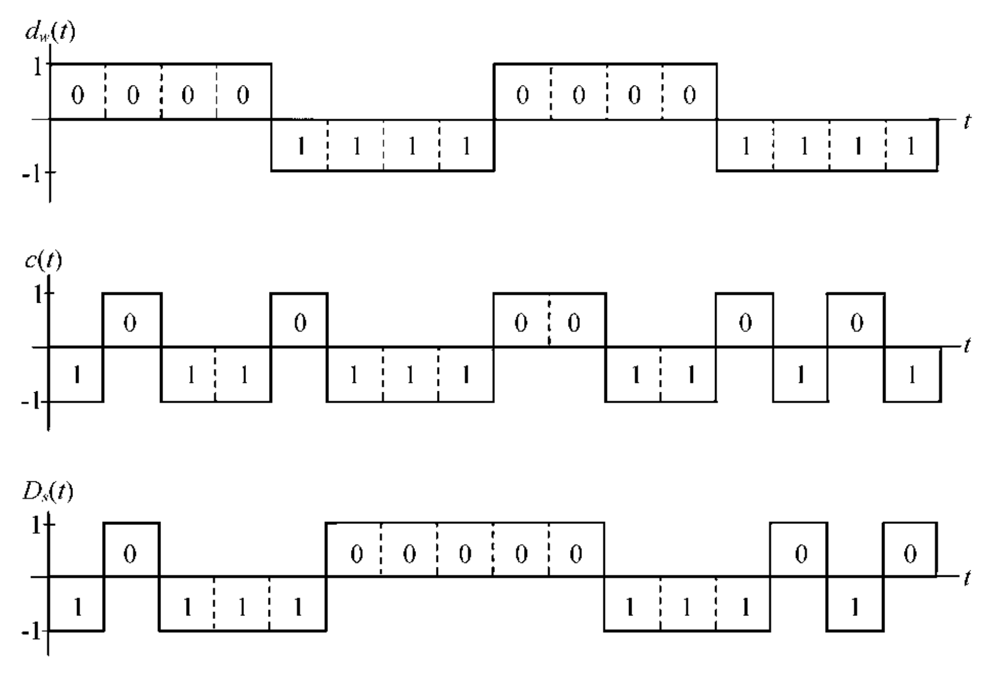
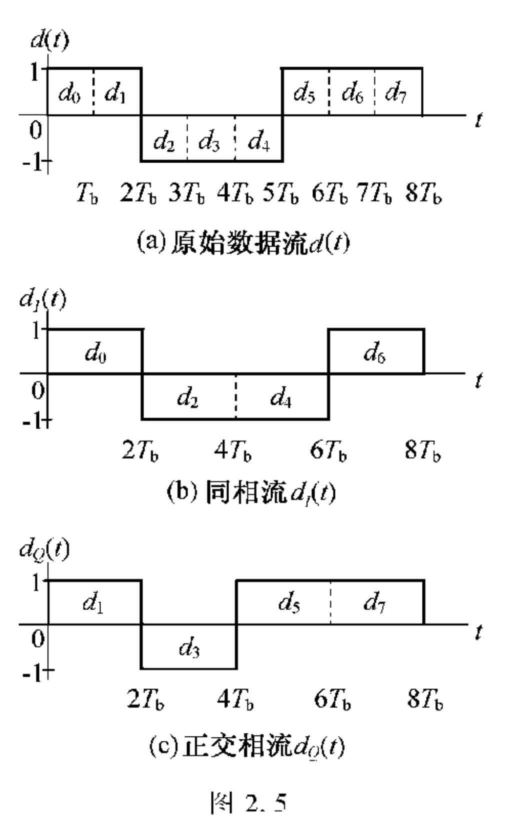


图 2.4　由长码 PN 码片 $c(t)$ 相乘的直接序列扩展 $d_w(t)$

2.5　QPSK 和偏移 QPSK 调制

为了使频带效率最大，高频谱效率的 CDMA 信道调制技术要求在相位正交的两个载波同时发送。正交调制在扩频中极为重要，它对某些类型的干扰不敏感。

令 $d(t)=d_0,d_1,d_2,\cdots$ 为原始数据流，$d(t)$ 为双极性脉冲，-1 代表二进制 1. *TIF*; *E* $+1$ 代表二进制 0，如图 2.5(a) 所示。该脉冲数据流分成同相流 $d_I(t)=d_0,d_2,d_4,\cdots$(偶数比特)和正交相位流 $d_Q(t)=d_1,d_3,d_5,\cdots$(奇数比特)，如图 2.5(b) 和图 2.5(c) 所示。注意，$d_I(t)$ 和 $d_Q(t)$ 分别具有 $d(t)$ 的一半速率。



(a) 原始数据流 $d(t)$

(b) 同相流 $d_I(t)$

(c) 正交相流 $d_Q(t)$

图 2.5

将每个 $d_I(t)$ 和 $d_Q(t)$ 幅度调制到载波的余弦和正弦函数上，可以得到 QPSK 波形，如图 2.6 所示。

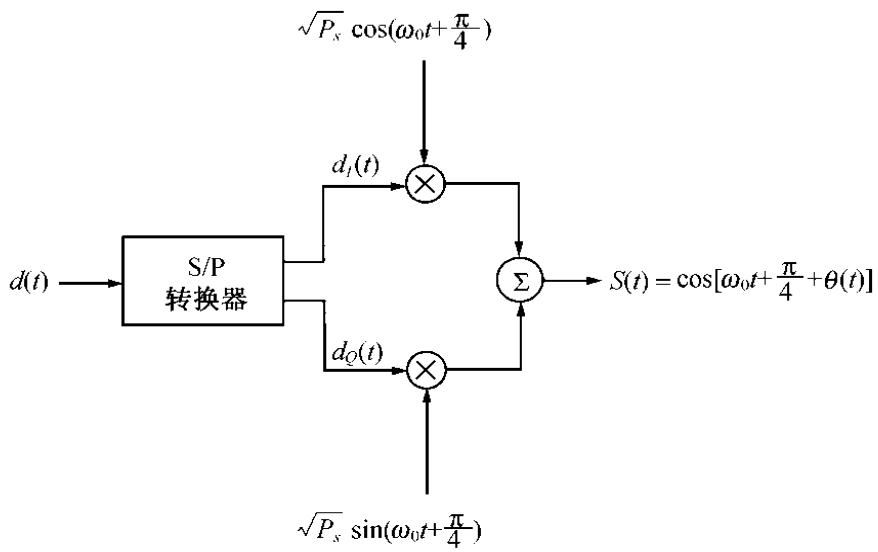


图 2.6 QPSK 扩频调制器

$$
\begin{aligned}
s(t) &= \sqrt{P_s}d_I(t)\cos(\omega_0 t+\frac{\pi}{4})+\sqrt{P_s}d_Q(t)\sin(\omega_0 t+\frac{\pi}{4}) \\
&= \sqrt{2P_s}(\cos(\omega_0 t+\frac{\pi}{4})\cos\theta(t)-\sin(\omega_0 t+\frac{\pi}{4})\sin\theta(t)) \\
&= \sqrt{2P_s}\cos[\omega_0 t+\frac{\pi}{4}+\theta(t)]
\end{aligned}
\tag{2.7}
$$

其中，$\cos\theta(t)=d_I(t)/\sqrt{2}$，$\sin\theta(t)=-d_Q(t)/\sqrt{2}$，$\theta(t)=-\arctan(d_Q(t)/d_I(t))$。

同相流 $d_I(t)$ 以 $+1$（二进制 0）或 -1（二进制 1）对余弦函数进行幅度调制，产生一个BPSK波形。同样地，正交相位流 $d_Q(t)$ 对余弦函数进行调制，产生另一个与前面余弦函数产生波形正交的 BPSK 波形。这样，载波的这两个正交分量就产生 QPSK 波形。

由于 $d_I(t)$ 和 $d_Q(t)$ 这两个扩展波形只取值 ± 1，这些扩展波形被认为是码片级同步的，除非它们彼此相互完全独立。$\theta(t)$ 值对应于 $d_I(t)$ 和 $d_Q(t)$ 可能的四种组合之一。对应 $d_I(t)$ 和 $d_Q(t)$ 特定值的 QPSK 调制器输出 $s(t)$，相应于 $\theta(t)$ 的相位偏移可按如下方式确定：

1. 若 $\theta(t)=-\frac{\pi}{4}$，则有 $d_I(t)=\sqrt{2}\cos\left(-\frac{\pi}{4}\right)=1$ 和

   $d_Q(t)=-\sqrt{2}\sin\left(-\frac{\pi}{4}\right)=1$，故 $s(t)=\sqrt{2P_s}\cos\omega_0 t$.

2. 若 $\theta(t)=\frac{\pi}{4}$，则有 $d_I(t)=\sqrt{2}\cos\left(\frac{\pi}{4}\right)=1$ 和

   $d_Q(t)=-\sqrt{2}\sin\left(\frac{\pi}{4}\right)=-1$，故 $s(t)=-\sqrt{2P_s}\cos\omega_0 t$ （2.8）

3. 若 $\theta(t)=\frac{5\pi}{4}$，则有 $d_I(t)=\sqrt{2}\cos\left(\frac{5\pi}{4}\right)=-1$ 和

   $d_Q(t)=-\sqrt{2}\sin\left(\frac{5\pi}{4}\right)=1$，故 $s(t)=\sqrt{2P_s}\sin\omega_0 t$.

4. 若 $\theta(t)=\frac{3\pi}{4}$，则有 $d_I(t)=\sqrt{2}\cos\left(\frac{3\pi}{4}\right)=-1$ 和

   $d_Q(t)=-\sqrt{2}\sin\left(\frac{3\pi}{4}\right)=-1$，故 $s(t)=-\sqrt{2P_s}\cos\omega_0 t$.

这样，QPSK 信号空间如图 2.7 所示。

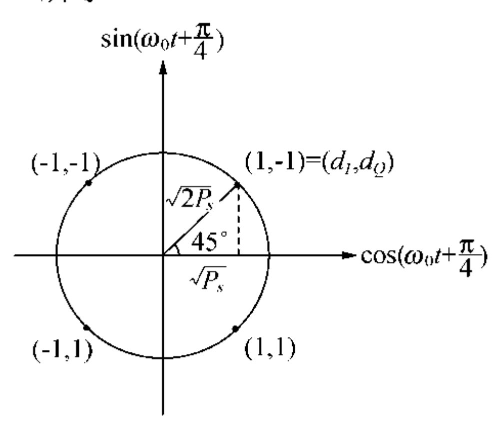
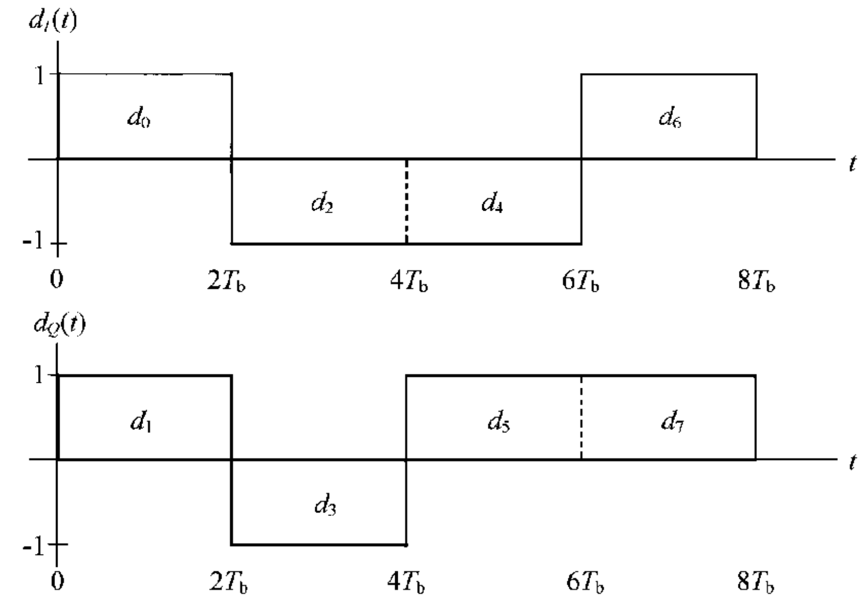


图 2.7　QPSK 信号空间

持续时间为 T_b 的原始数据流 $d(t)$ 可以通过串/并转换器将持续时间 $2T_b$ 划分为同相流 $d_I(t)$ 和正交相位流 $d_Q(t)$（参见图 2.6）。QPSK 调制时，同相和正交相位数据流都以 $1/2T_b$ b/s 速率发送并同步对齐，以使发送时间一致。非偏移 QPSK 调制时，两种脉冲流 $d_I(t)$ 和 $d_Q(t)$ 发送时间一致，如图 2.8 所示。由于 $d_I(t)$ 和 $d_Q(t)$ 一致对齐，所以载波相位每 $2T_b$ 秒变化一次相位，导致了图 2.7 所示的四种相位之一。



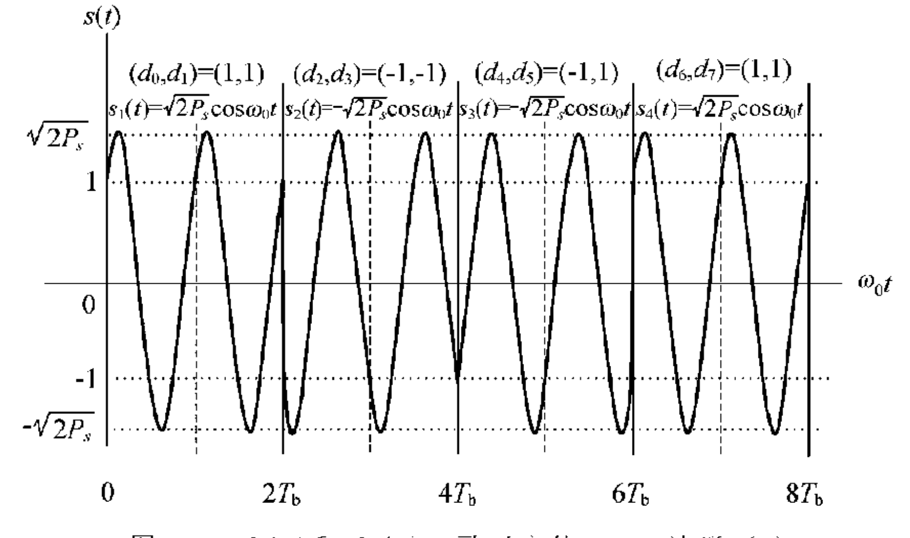
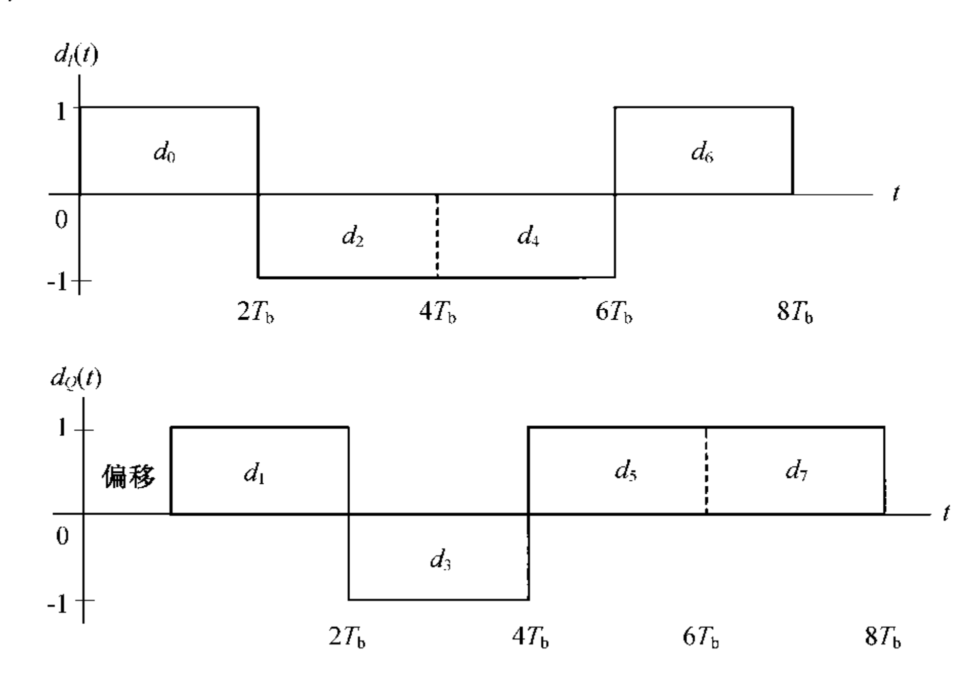
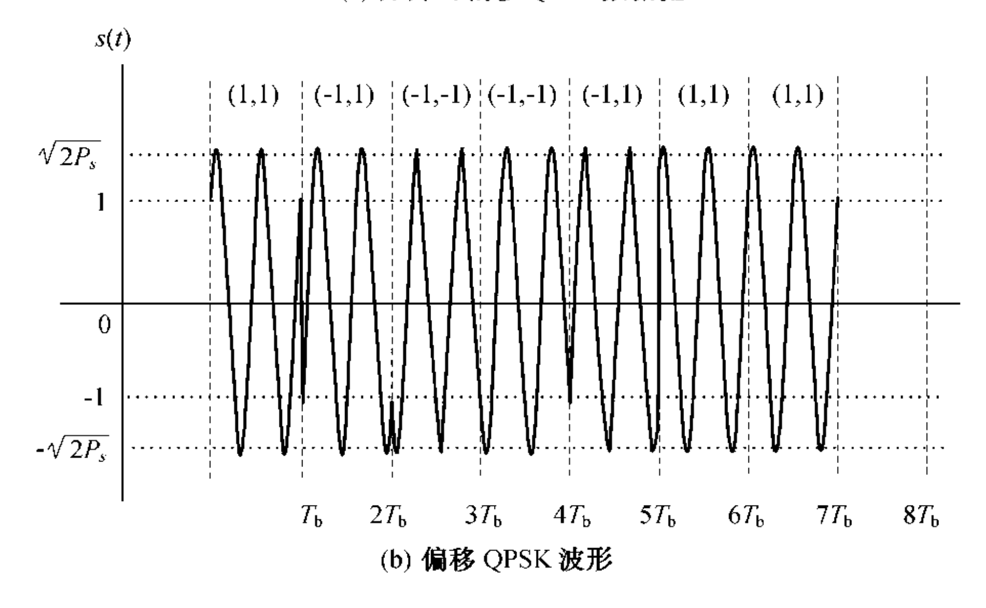


图 2.8　$d_I(t)$ 和 $d_Q(t)$ 一致对齐的 QPSK 波形 $s(t)$

方程(2.7)也用于偏移 QPSK(OQPSK)信令。OQPSK 和标准的非偏移 QPSK 在两个基带波形对齐方面不同。两种脉冲流 $d_I(t)$和 $d_Q(t)$相互交错,因此不是同时地改变状态。这两种调制方案的不同在于脉冲流 $d_I(t)$和 $d_Q(t)$定时移位了,以至于两种数据流差 T_b 对齐,如图 2.9 所示。载波相位每 T_b 秒可能的变化限制在 0°和 ±90°。然而在非偏移 QPSK 下,两种脉冲流 $d_I(t)$和 $d_Q(t)$时间一致,载波相位每 $2T_b$ 秒变化一次,如图 2.8 所示。对应于OQPSK数据流的典型 OQPSK 波形如图 2.9(b)所示。



(a) 分开的偏移 QPSK 数据流



(b) 偏移 QPSK 波形

图 2.9　$d_I(t)$和 $d_Q(t)$脉冲流交错的 OQPSK 波形

2.6　长码的产生

长码提供了有限的保密性。长码是 $2^{42}-1$ 的 PN 序列,用于前向 CDMA 信道的扰码和反向 CDMA 信道的扩频。长码在前向业务信道和反向业务信道惟一标识移动台。长码的特点是使用长码掩码来形成公用长码或私有长码。长码还用在相同 CDMA 信道上区分多重接入信道。

当在接入信道上传输时，先进行长码直接序列扩展，然后发送。扩展操作包括 64 阶正交调制器输出序列和长码的模 2 加，如图 2.10 所示。

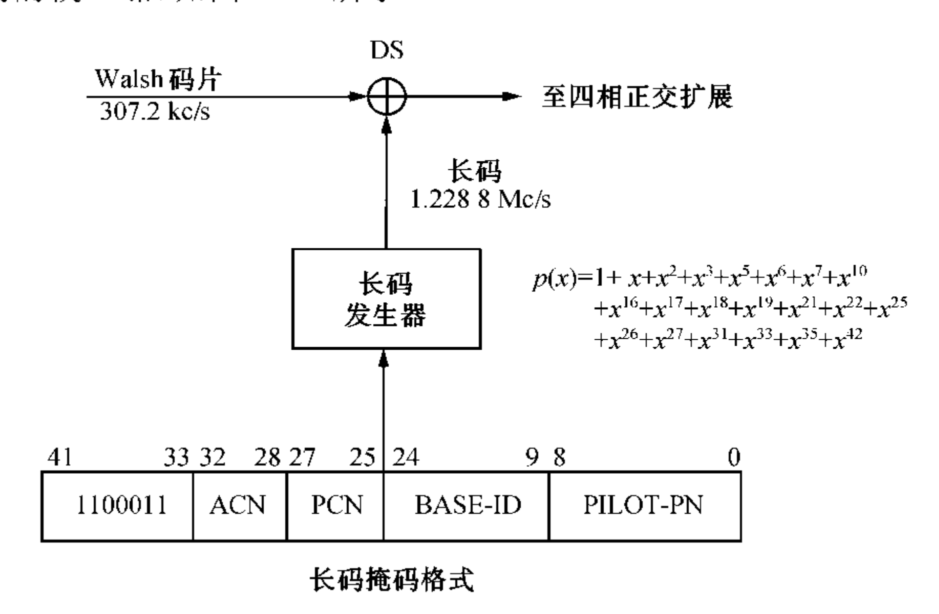


图 2.10　接入信道长码

长码周期为 $2^{42}-1$ 个码片，由码发生器的 LFSR 抽头多项式 $p(x)$ 指定：

$$p(x)=1+x+x^2+x^3+x^5+x^6+x^7+x^{10}+x^{16}+x^{17}+x^{18}+x^{19}+x^{21}+x^{22}+x^{25}+x^{26}+x^{27}+x^{31}+x^{33}+x^{35}+x^{42} \tag{2.9}$$

对所有 42 位掩码和 LFSR 42 级输出的“与”结果进行“异或”操作可以得到长码的每个 PN 码片，如图 2.11 所示。

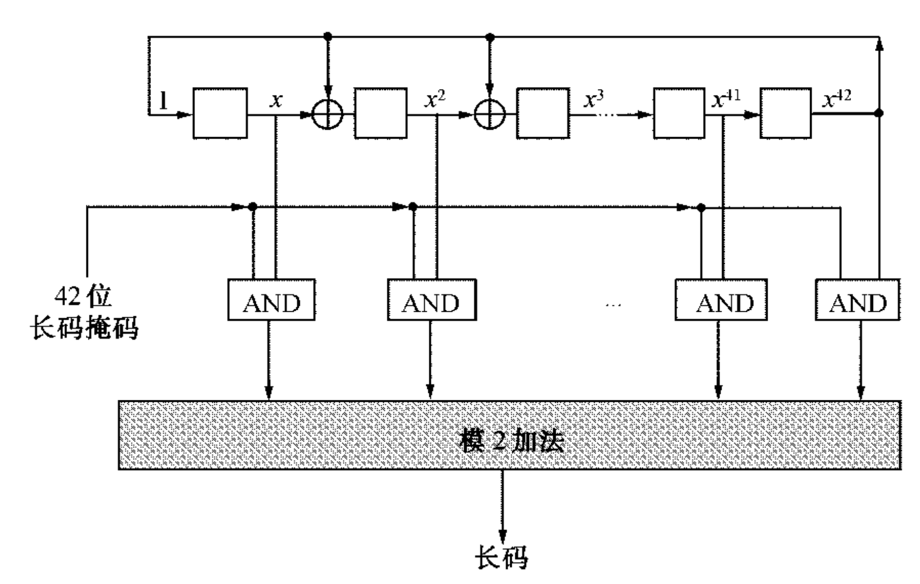


图 2.11　长码发生器

长码掩码包括42位二进制序列，它是长码的惟一标识。长码掩码根据移动台传输的信道类型而不同。图2.11显示了长码发生器产生的PN码片(1.228■8 kc/s)，它由长码掩码激活。

2.7　数据扰码

数据扰码技术用在前向业务信道和寻呼信道。数据扰码在块交织器输出端以19.2 ks/s速率进行。数据扰码通过对块交织器输出数据和长码PN码片的量化抽样值(0或1)进行模2加实现。该PN序列等价于以1.228■8 MHz时钟速率的长码操作，只有每64比特输出1比特的第1比特用于19.2 ks/s的数据扰码。

前向业务信道数据使用长码发生器进行扰码。图2.12显示了数据扰码机制。以1.228■8 MHz时钟速率进行的长码操作将产生PN码片序列，作为长码发生器的输出。

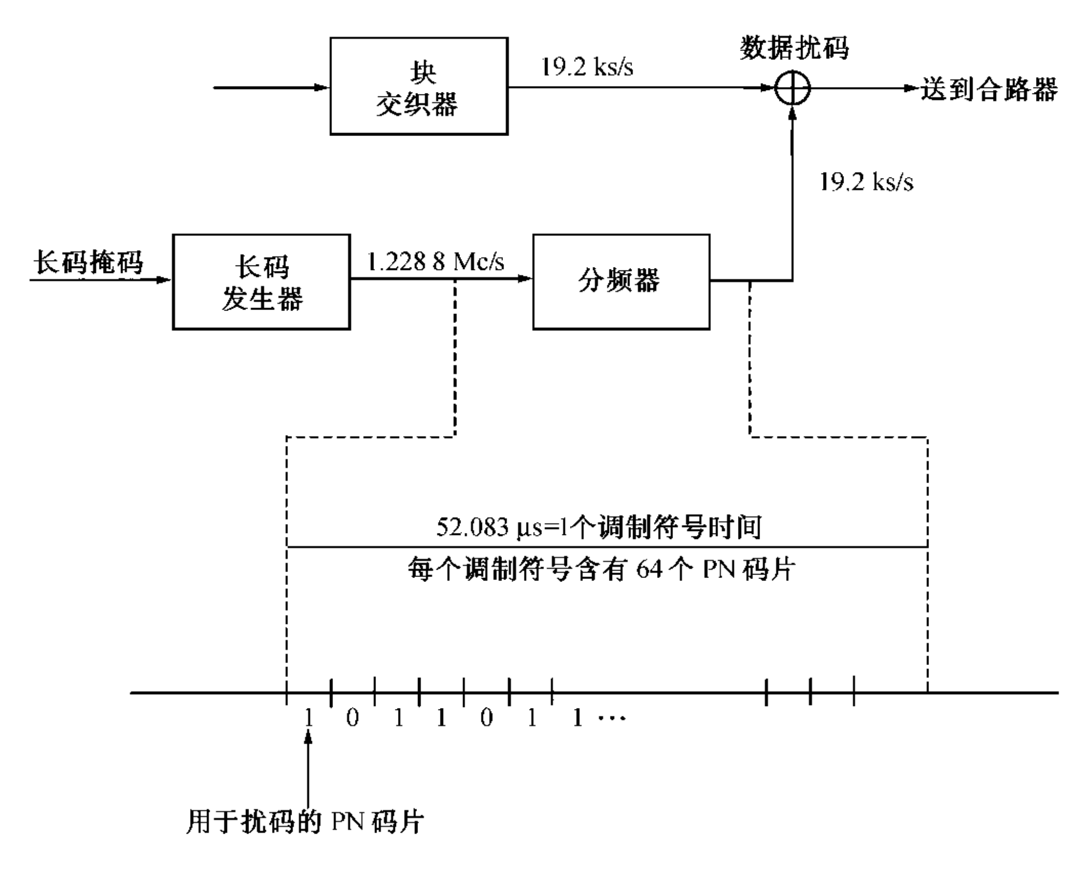


图2.12　数据扰码机制

当长码分成64位长的小块时，每64比特的第1位比特是量化抽样值，这些均匀量化值用于19.2 ks/s的数据扰码。量化器的功能是通过每64位输出1位来减小长码的长度。

CDMA系统中话音保密是通过用于PN扩展的私有长码掩码提供的，只在业务信道提供话音保密控制。所有呼叫使用PN扩展的公用长码掩码初始化。如果没有进行鉴权就不能转向使用私有长码掩码。要转向私有或公用码掩码，基站或移动台在业务信道发送长码转换要求的指令。移动台或基站分别操作响应该指令。基站同时还发送含有适当设置的PRIVATE-LCN位的切换指令消息，以进行向公用长码掩码的转换。

2.8　CDMA 码道正交相位扩展

直接序列扩展之后，反向业务信道和接入信道进行正交相位扩展。

另外，前向 CDMA 信道正交扩展之后，每个码道（导频、同步、寻呼或前向业务信道）进行正交相位扩展。

扩展序列是长度为 2^{15} 或 32 768 个 PN 码片的正交相位序列。该正交相位序列称为导频 PN 序列，分别基于下列抽头多项式：

对于同相 I 序列

$$P_I(x) = 1 + x^5 + x^7 + x^8 + x^9 + x^{13} + x^{15} \tag{2.10}$$

对于正交相位 Q 序列

$$P_Q(x) = 1 + x^3 + x^4 + x^5 + x^6 + x^{10} + x^{11} + x^{12} + x^{15} \tag{2.11}$$

通过下列互逆多项式获得 I 相和 Q 相导频 PN 序列：

$$i(x) = x^{15} P_I(x^{-1}) = 1 + x^2 + x^6 + x^7 + x^8 + x^{10} + x^{15} \tag{2.12}$$

和

$$q(x) = x^{15} P_Q(x^{-1}) = 1 + x^3 + x^4 + x^5 + x^9 + x^{10} + x^{11} + x^{12} + x^{15} \tag{2.13}$$

基于方程 2.12 和 2.13 的 LFSR（左移寄存器）最大长度为 $2^{15}-1$，可以通过下列线性递归操作产生：

$$i(n) = i(n-15) \oplus i(n-10) \oplus i(n-8) \oplus i(n-7) \oplus i(n-6) \oplus i(n-2) \tag{2.14}$$

和

$$q(n) = q(n-15) \oplus q(n-12) \oplus q(n-11) \oplus q(n-10) \oplus q(n-9) \oplus q(n-5) \oplus q(n-4) \oplus q(n-3) \tag{2.15}$$

其中 $i(n)$ 和 $q(n)$，$1 \leqslant n \leqslant 32\,767$ 是二进制 0 或 1，$i(15)=1$，$q(15)=1$，$\oplus$ 表示模 2 加。

I 和 Q 导频序列（$\overline{P}_I$ 和 $\overline{P}_Q$）由 $i(n)$ 和 $q(n)$ 产生，LFSR 初始内容为：IC =（1000000000000000）。$\overline{P}_I$ 和 $\overline{P}_Q$ 可以使用长度为 $2^{15}-1$ 的多项式 $i(n)$ 和 $q(n)$ 计算，即：

1. 使用 $i(n) = i(n-15) = 0, 1 \leqslant n \leqslant 14, i(15) = 1, i(n) = \sum_{k=15,10,8,7,6,2} i(n-k), 16 \leqslant n \leqslant 32\,767$ 首先计算 $i(n)$，连续 14 个 0 之后，然后在 $i(n)$ 处插入 0，可以得到同相导频 PN 序列。
2. 类似地，使用 $q(n) = q(n-15) = 0, 1 \leqslant n \leqslant 14, q(15) = 1, q(n) = \sum_{k=15,12,11,10,9,5,4,3} q(n-k), 16 \leqslant n \leqslant 32\,767$，在 $q(n)$ 的第 15 位插入 0，可以得到正交相位导频 PN 序列 $\overline{P}_Q$。

2.9　全部码道的正交信道化

对于 W-CDMA 系统，反向信息信道和反向业务信道的信令信道都是经过导频码序列、业务码 I 序列、业务码 Q 序列和信令码序列直接序列扩展后再发送。这些码序列由一个周期为81 920码片的码序列产生，经过长码序列发生器生成数据和适当的 Walsh 函数的模 2 加实现，如图 2.13 所示。在图 2.13（b）中，指数号是 0 的 Walsh 函数分配给导频信道，指数号是 1 的 Walsh 函数分配给业务 I 和 Q 以及接入代码。指数号是 2 的 Walsh 函数分配给信令信道。每个码序列导致了周期为 20

ms 的 81 920 码片。图 2.13(a)中显示了长码 PN 码的产生。

长码的生成种子来自于基站发送的寻呼信道消息中。该长码是周期为 $2^{23}-1$ 码片的截短线性码，满足下列多项式确定的线性递归：

$$p(x)=1+x+x^{2}+x^{22}+x^{32} \tag{2.16}$$

固定码片速率为 4.096 Mc/s 的 Walsh 函数在反向业务信道或接入信道上(参见图 2.13(b))为全部信道(反向导频信道、反向信息信道和反向信令信道)提供了正交信道化。都使用表 2.7 定义的三个 64 阶 Walsh 函数。

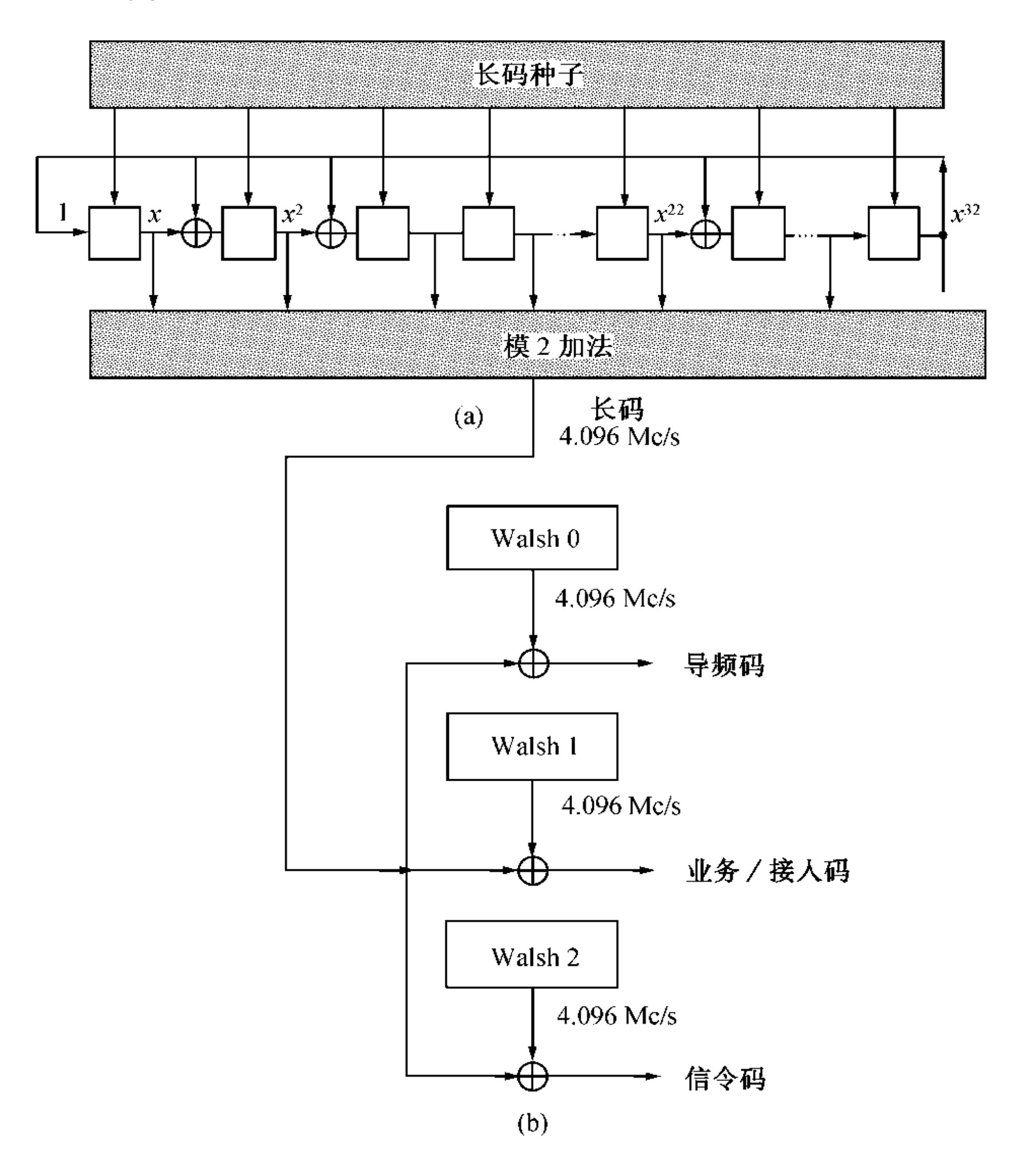


图 2.13 全部码道的正交信道化

(a) 长码的 PN 码发生器 (b) 长码的 PN 序列发生器

表2.7 反向链路 Walsh 函数

(一个 Walsh 函数中的 Walsh 码片)

Walsh 索引	 0123	 4567	11 8901	1111 2345	1111 6789	2222 0123	2222 4567	2233 8901	3333 2345	3333 6789	4444 0123	4444 4567	4455 8901	5555 2345	5555 6789	6666 0123
0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
1	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101
2	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011

Walsh 函数的时间对齐如下：表 2.7 列标题中 0 表示的第一个 Walsh 码片开始于每一帧的 5ms 长度内的第一个码片。

例 2.4　参考图 2.13(a)PN 码发生器的抽头 $p(x)=1+x+x^2+x^{22}+x^{32}$，其向量表示为：1110000000 0000000000 0010000000 01；初始生成向量假设为：1001001100 1010011011 0101111000 11。利用这两个向量，可产生如表 2.8 所示的长码序列。

表 2.8　使用 PN 码发生器的长码计算

移位编号	寄存器内容	1 的个数	长码比特
0	10010011001010011011010111100011	17	1
1	10101001100101001101100011110001	16	0
2	10110100110010100110111001111000	17	1
3	01011010011001010011011100111100	17	1
4	00101101001100101001101110011110	17	1
5	00010110100110010100110111001111	17	1
6	11101011010011001010010011100111	18	0
7	10010101101001100101000001110011	15	1
8	10101010110100110010101000111001	16	0
9	10110101011010011001011100011100	17	1
10	01011010101101001100101110001110	17	1

这样，产生的长码序列可以如下表示：

1011110101110100 0010111010110010 1001111011101011 0111001111000001
0010100011011010 0110110111011100 1010110100101100 0100101000000101
1100001010110101 0101101101001111 1001111011101101 0011101110100000
0011000010001111 1000000110000000 1111001110010010 0110000000011010
1101110001001000 0011001001001000 1000011111011001 0110100110100001
1111111101110001 1100000000010101 1111100100101011 0000101010101011
0100010100000000 0111010101001101 1000100001100011 0101001000110100
0011010000100101 1011011111100001 0111010010100101 0010111011101101
1111010000000110 1100100100110011 1000000011110000 0100001110001010
0110000001001001 11011000

例 2.5　如图 2.13(b)所示，导频码、业务/接入码和信令码都是由长码序列和适当的 Walsh 函数模 2 加得到的。使用三个 64 阶 Walsh 函数(参见表 2.7)，这些码序列产生了周期为 20 ms 的 81 920 个码片的码序列。

1. 导频码产生

指数号为 0 的 Walsh 函数 W0 分配到导频码。

导频码 = 长码⊕Walsh0

长码：　1011110101110100 0010111010110010 1001111011101011 01110011100001
⊕
Walsh0：000000000000000 000000000000000 000000000000000 00000000000000
导频码：1011110101110100 0010111010110010 1001111011101011 01110011100001

该例计算仅使用长码序列的第一行。

这样，导频码就由长码序列和指数号为 0 的 Walsh0 函数模 2 加产生：

1011110101110100 0010111010110010 1001111011101011 0111001111000001
0010100011011010 0110110111011100 1010110100101100 0100101000000101
1100001010110101 0101101101001111 1001111011101101 0011101110100000
0011000010001111 1000000110000000 1111001110010010 0110000000011010
1101110001001000 0011001001001000 1000011111011001 0110100110100001
1111111101110001 1100000000010101 1111100100101011 0000101010101011
0100010100000000 0111010101001101 1000100001100011 0101001000110100
0011010000100101 1011011111100001 0111010010100101 0010111011101101

2. 业务/接入码产生

指数号为 1 的 Walsh 函数 W1 分配到业务/接入码。

业务/接入码 = 长码⊕Walsh1

长码： 1011110101110100 0010111010110010 1001111011101011 0111001111000001

⊕

Walsh1： 0101010101010101 0101010101010101 0101010101010101 0101010101010101

业务/接入码：1110100000100001 0111101111100111 1100101110111110 0010011010010100

业务/接入码由长码序列和指数号为 1 的 Walsh 函数模 2 加得到：

1110100000100001 0111101111100111 1100101110111110 0010011010010100
0111110110001111 0011100010001001 1111100001111001 0001111101010000
1001011111100000 0000111000011010 1100101110111000 0110111011110101
0110010111011010 1101010011010101 1010011011000111 0011010101001111
1000100100011101 0110011100011101 1101001010001100 0011110011110100
1010101000100100 1001010101000000 1010110001111110 0101111111111110
0001000001010101 0010000000011000 1101110100110110 0000011101100001
0110000101110000 1110001010110100 0010000111110000 0111101110111000

3. 信令码产生

指数号为 2 的 Walsh 函数 W2 分配到信令码。

信令码 = 长码⊕Walsh2

长码： 1011110101110100 0010111010110010 1001111011101011 0111001111000001

⊕

Walsh2：0011001100110011 0011001100110011 0011001100110011 0011001100110011

信令码：1000111001000111 0001110110000001 1010110111011000 0100000011110010

信令码码由长码序列和指数号为 2 的 Walsh 函数模 2 加得到：

1000111001000111 0001110110000001 1010110111011000 0100000011110010
0001101111101001 0101111011101111 1001111000011111 0111100100110110
1111000110000110 0110100001111100 1010110111011110 0000100010010011
0000001110111100 1011001010110011 1100000010100001 0101001100101001
1110111101111011 0000000101111011 1011010011101010 0101101010010010

1100110001000010　1111001100100110　1100101000011000　0011100110011000
0111011000110011　0100011001111110　1011101101010000　0110000100000111
0000011100010110　1000010011010010　0100011110010110　0001110111011110

发送之前，反向信息信道和信令信道经过业务码 I 序列、业务码 Q 序列和信令码序列直接序列扩展。

在反向 W-CDMA 信道上发送的数据封装成 5 ms 帧。以 64 kb/s，32 kb/s 和 16 kb/s 发送的反向信息信道帧分别包含 320，160 和 80 比特数据。

2.10　鉴权和加密

移动台必须和基站联合进行移动台鉴权。鉴权是这样一个过程，即信息在移动台和基站之间交互，用于确认移动台身份。只有当能够表明移动台和基站处理的 SSD（共享加密数据）集是相同的时候，鉴权过程才能成功。例如，在 CDMA 鉴权协议中，移动台和基站互相都有匹配的加密认证符（即相同的 SSD），实际上是由认证算法产生和发布的符号的短消息摘要。

2.10.1　鉴权

共享加密数据（SSD）是移动台半永久存储器中存储的 128 比特长的数据结构，很容易为基站所用。SSD 分为两个不同子集，每个子集用于支持不同过程：SSD-A 用于支持鉴权过程，SSD-B 用于支持话音保密和消息加密。

在计算机通信网中，常常有必要验证通信一方的身份。一种实用的方法就是使用单向 Hash 函数的加密认证协议。CDMA 系统中的鉴权通过在移动台和基站之间相互交换信息来确认移动台身份的过程。

考虑一种可能的鉴权方案用于分组密码。可能使用对称分组密码算法（如 DES）来计算 18 位的 Hash 值。如果分组算法安全，那么可以假设单向 Hash 函数也是安全的。

鉴权算法设备输入 152 位消息分组 M，利用 DES，M 首先分成 64 位长的块，使得 $M = M_1, M_2, M_3, \cdots$。第一个消息块 M_1 成为 DES 密钥。通过对 152 位消息填充 40 位 0 从而将它映射成 192 个比特值，便可完成 64 位消息块的划分。适当的填充是为了方便将消息划分成某些固定长的块。这里的鉴权方案就是要从 192 位长的消息中产生 18 位鉴权数据。

这 18 位用于鉴权数据的消息摘要（AUTHR）也可以通过添加 24 位填充而得到的 176 个比特消息进行 Hash 变换而推导。

例 2.6　图 2.15 是一个鉴权计算的典型例子。首先，将 152 位消息填充 40 位扩充成 192 位消息。其次，将填充的 192 位消息分成三块：$M_1 = 64$ 位，$M_2 = 64$ 位，$M_3 = 64$ 位。M_1 成为 64 位密钥，如图 2.14 所示。第一个消息块 M_1 可用于图 2.15 中的灰色模块，可被 Hash 成 48 位加密密钥。8.11.2 节将详细讨论这个问题。

有很多方法用于鉴权数据（AUTHR）计算，第 8 章将加以描述。

为了计算上面描述的 18 位鉴权数据，考虑两种情况：（1）176 位已填充的消息（152 位消息加上 24 位填充位），（2）添加 40 位填充位的 192 位消息。

除了这些，还有一些计算 18 位消息摘要的技术，包括 DM 方案。要进行 Hash 变换的消息首先

划分成定长的块：$M_1, M_2, \cdots, M_t$。消息块 $M_i, 1 \leqslant i \leqslant t$ 将作为加密密钥。前面一个消息块使用该密钥加密，然后自身进行异或运算。得到的结果作为下一轮的输入。这样，在最后一轮，消息长度将变为 M_t。DM 方案使用 DES 分组密码来构造 Hash 函数，使得 $\boldsymbol{H}_i = E_{M_i}(\boldsymbol{H}_{i-1}) \oplus \boldsymbol{H}_{i-1}, 1 \leqslant i \leqslant t$，其中 $\boldsymbol{H}_0$ 是初始向量。

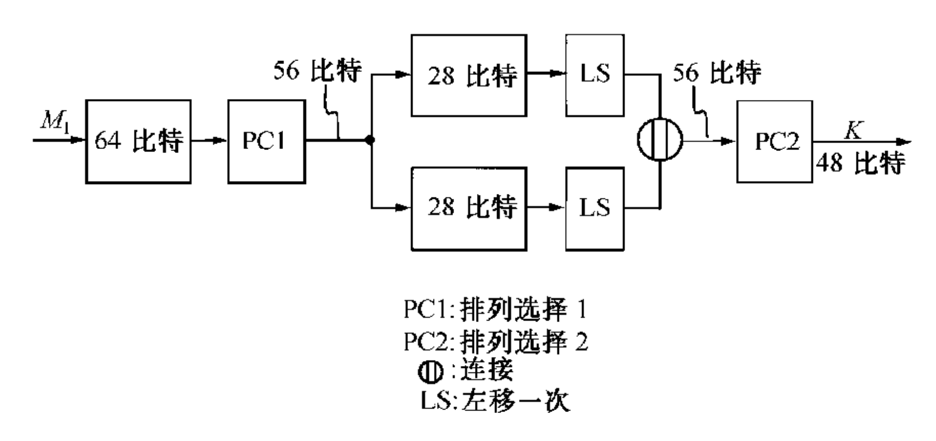


图 2.14 基于 *DES* 调度的密钥产生

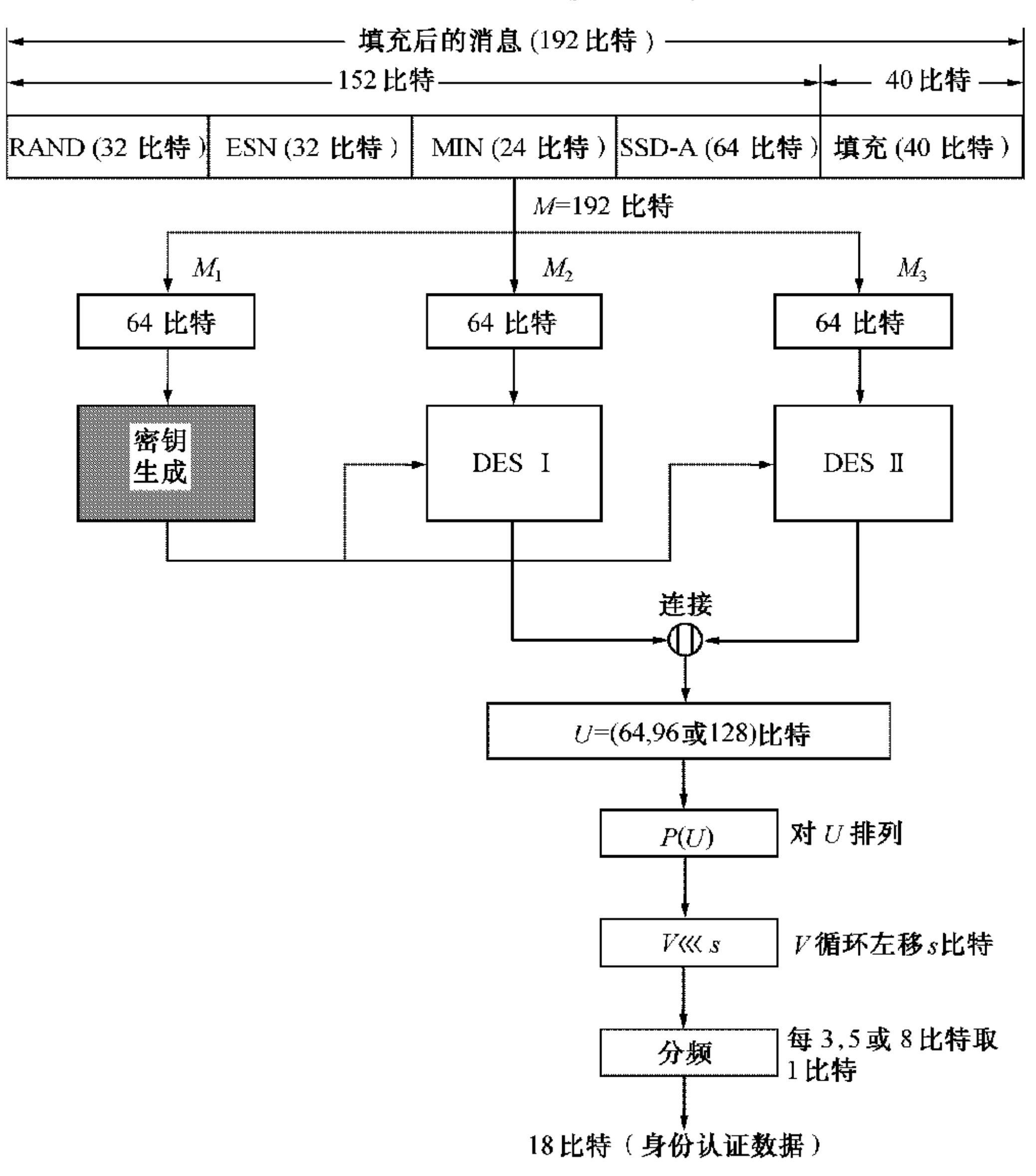


图 2.15 移动台注册、惟一查询响应过程、发起或终止而进行的身份认证计算

另一种方案是改进的 CBC 模式。改进的 CBC 方案加密步骤描述如下：

$$\boldsymbol{H}_1 = E_{H_0 \oplus M_1}(\boldsymbol{H}_0) \oplus \boldsymbol{H}_0$$

$$\boldsymbol{H}_2 = E_{H_1 \oplus M_2}(\boldsymbol{H}_1) \oplus \boldsymbol{H}_1$$

$$\boldsymbol{H}_3 = E_{H_2 \oplus M_3}(\boldsymbol{H}_2) \oplus \boldsymbol{H}_2$$

其中 $\boldsymbol{H}_0$ 是初始向量，$\boldsymbol{H}_i \oplus M_j$ 表示解密密钥。

MD5 算法用于计算 SSD-A 和 SSD-B。MD5 利用任意长度消息产生 128 位消息摘要输出。MD5 将输入消息在可以分为 16 个 32 位的子块的 512 位长的块内处理。消息摘要是一组 32 位长的块，将它们连接起来就成为单一的 128 位长的 Hash 码。由于 SSD 更新由存储在移动台的 128 位长数据完成，它被分为两个不同的 64 位子状态，即 SSD-A-NEW 和 SSD-B-NEW。

2.10.2　消息加密和信息安全

为了增强鉴权过程和保护敏感用户信息（例如 PIN），提供了一种方法对选定业务信道信令消息的某些特定字段加密。然而，TIA/EIA/IS-95 都没有讨论和列出需要加密的消息和字段，原因是加密算法的使用完全受控于美国国际交易和武器规则（ITAR）和出口管理规则。

如果鉴权没有执行，就不对消息加密（即接入参数消息的 AUTH 字段等于 00）。信令消息加密是对每个呼叫单独控制的。呼叫的初始加密模式在信道分配时由加密消息的信令加密字段值给定。如果信令加密字段设为 00，则不对消息加密。

要在信道分配之后启用加密，基站将信令字段设为 01 来发送加密消息。

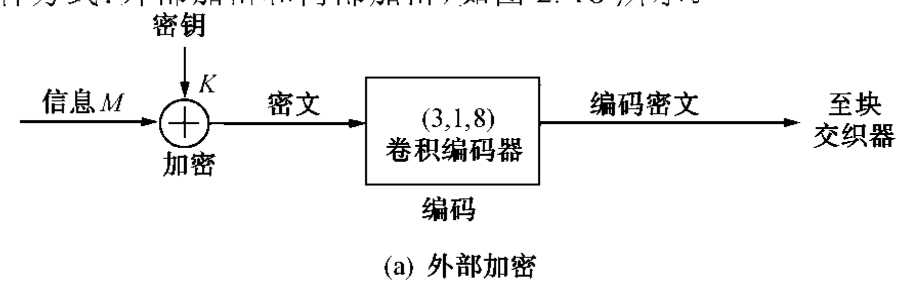
要禁用信令消息加密，基站将信令字段设为 00 来发送加密消息。

每路反向业务信道消息包含一个加密字段来识别消息创建时激活的消息加密模式。

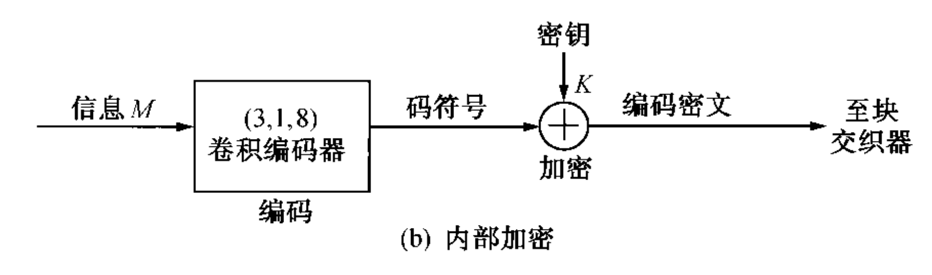
反向业务信道上发送的数据封装成 20 ms 帧。数据帧可以用 9 600 b/s，4 800 b/s，2 400 b/s 或 1 200 b/s 的变速率在反向业务信道发送。反向业务信道用于在呼叫过程向基站传输用户数据和信令信息。

每个数据帧持续时间为 20 ms，根据发送速率确定包含信息 CRC 尾比特，或信息和尾比特。例如，对于 4 800 b/s 的速率，96 比特帧包含 80 位信息、8 位 CRC 和 8 位尾比特。

消息加密有两种方式：外部加密和内部加密，如图 2.16 所示。



(a) 外部加密



(b) 内部加密

图 2.16　消息加密方案

消息信息 M 首先由密钥流 K 加密，然后使用(3, 1, 8)卷积编码器编码。这种加密方案称为“外部加密”。第二种方案如图 2.16(b)所示，编码在加密之前进行，解密在解码之前，这种加密方案称为“内部加密”。

例 2.7　考虑外部加密情况，假设 4■800 b/s 帧(20 ms)的 80 位信息数据如下：

1010110011 1001101111 0010010100 0100110011

0011001000 1110010110 1001100110 0110011110

CRC 计算使用下列生成多项式：

$g(x)=1+x+x^3+x^4+x^7+x^8$，寄存器初始状态是全 1。CRC 比特计算为：

CRC = 10011101

这样，将编码器尾比特(全 0)和上面的 CRC 以及信息数据连接就可得到下列 4 800 b/s 的 96 位帧数据：

1010110011 1001101111 0010010100 0100110011 0011001000

1110010110 1001100110 0110011110 1001110100 000000

这就是将要用加密密钥加密的数据序列，假设如下：

0100010111 1010101100 1010110011 0111000100 1010100110

1010010011 1100110101 0011100100 1100000010 100110

将信息帧数据和密钥序列做“异或”运算将得到下列加密文本数据序列：

1110100100 0011000011 1000100111 0011110111 100110110

0100000101 0101010011 0101111010 0101110110 100110

这就是要输入(3, 1, 8)卷积编码器的加密帧数据。

第3章　反向CDMA信道

CDMA(码分多址)是一种利用惟一码序列进行扩频多址接入数字通信的技术。CDMA 信道包括基站和移动台之间的前向 CDMA 信道和反向 CDMA 信道。前者是从基站到移动台的信道,后者是从移动台到基站的信道。

反向 CDMA 信道被移动台用来和基站通信,同时在发送之前通过直接序列扩频共享相同的 CDMA 频率分配。反向 CDMA 信道是从移动台到基站的反向链路。在反向 CDMA 信道发送的数据被封装成 20 ms 帧。反向 CDMA 信道包括接入信道和反向业务信道。接入信道用于短信令消息交换,提供呼叫发起、寻呼响应、指令和注册。反向业务信道用于从单个移动台向单个或多个基站传输用户数据和信令业务。图 3.1 显示了基站在逻辑反向 CDMA 信道接收到的所有移动台传输总和。

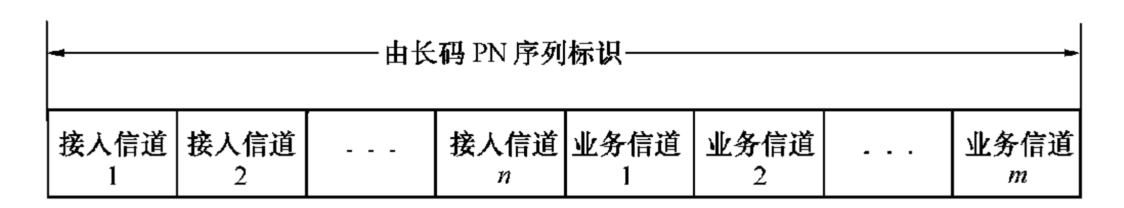


图 3.1　基站接收到的反向 CDMA 码道示例(1.23 MHz 信道)

3.1　接入信道

这部分内容描述移动台在接入信道发送的消息。图 3.2 显示了接入信道的完整结构。接入信道是反向 CDMA 码道,移动台用接入信道来发起和基站的通信并响应寻呼信道消息。每个寻呼信道和一个或多个接入信道对应。接入信道的数据率固定为 4 800 b/s。接入信道帧持续时间长 20 ms,只有当系统时间是 20 ms 的整数倍时才开始发送。接入信道传输包括接入信道报头(前同步)和接入信道消息封装。如图 3.2 所示,接入信道传输是经过编码、交织和调制的扩频信号。每路接入信道由不同的长 PN 码区分。

3.1.1　接入信道帧结构

每路接入信道包含 96 比特(20 ms 帧,数据率为 4 800 b/s)。每路接入信道帧格式包含 88 位信息比特和 8 位编码尾比特,如图 3.3 所示。

例 3.1　考虑向信息比特加上 8 位尾比特以增加数据速率。由于 8 位尾比特是加在 88 位信息比特后的,接入信道将每帧包含 96 比特,持续时间 20 ms。这样,数据速率 4.4 kb/s 将增加到 $96/20\times10^{-3}=4\ 800$ b/s。因此,任何接入信道的数据率固定在 4.8 kb/s。另外,对于每帧 88 比特信息,可以得到 $4400/88=50$ b/s,等价于 1 比特/帧,于是增加 8 位尾比特就等于 $50\times8=0.4$ kb/s。这证明数据速率增加到 $4.4+0.4=4.8$ kb/s。

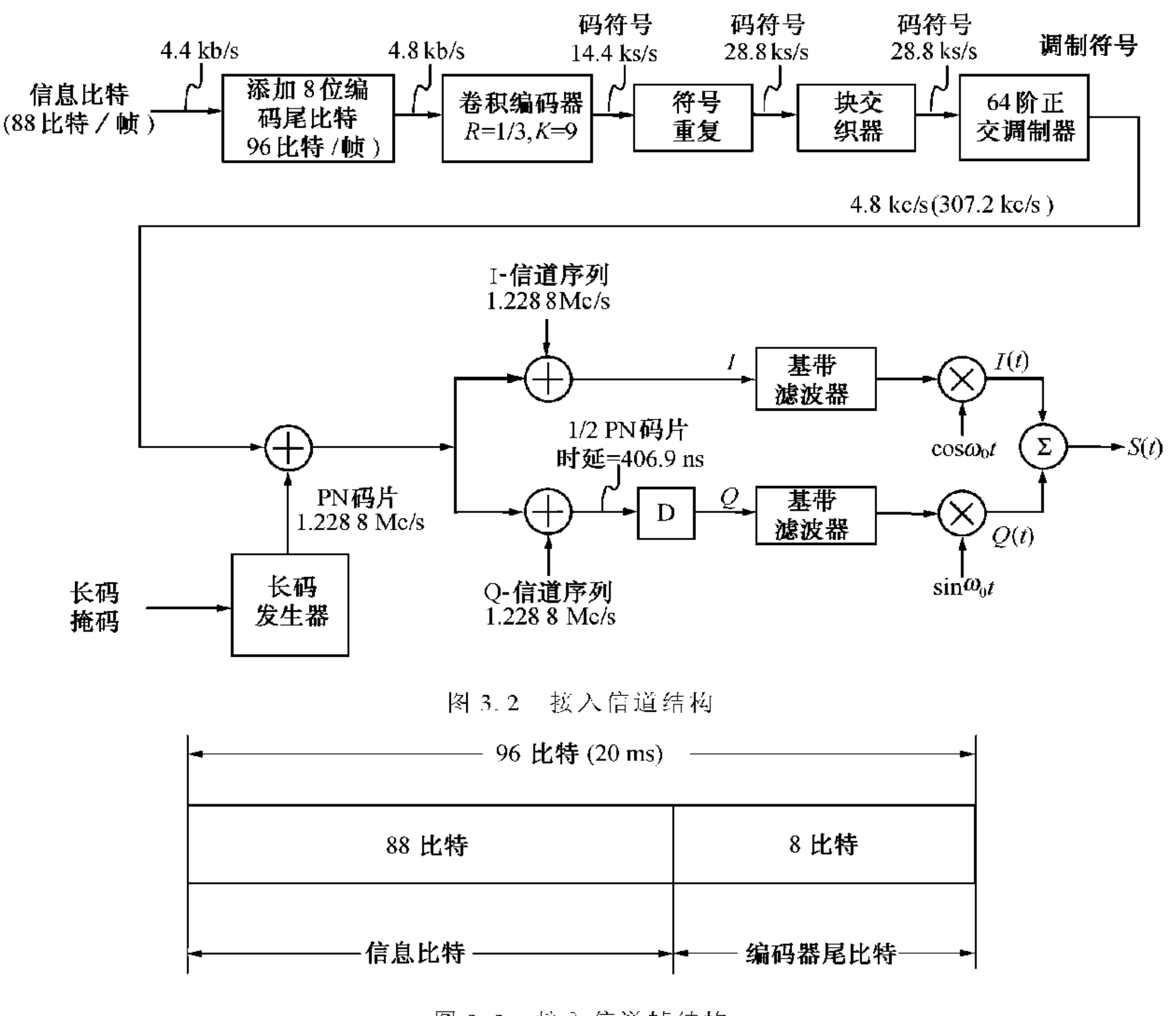


图 3.2 接入信道结构

图 3.3 接入信道帧结构

3.1.2 卷积编码

卷积码(n,k,m)表示码率 $R=k/n$，编码器级数 $m=K-1$，其中 K 是码约束长度。

移动台对在接入信道发送的数据卷积编码之后，再进行块交织。反向 CDMA 信道使用(3,1,8)卷积码，码率为 $R=1/3$，约束长度为 9。令 $g_k^{(n)}$ 表示生成序列，其中 $k=1$，$n=1,2,3$。于是该卷积码的生成序列为：$g_1^{(1)}=557$（八进制）=（101101111）（二进制），$g_1^{(2)}=663$（八进制）=（110110011）（二进制），$g_1^{(3)}=711$（八进制）=（111001001）（二进制）。由于 $k=1$，$n=3$，$m=8$，则该卷积码编码器包含单个输入端、一个 8 级移位寄存器、三个模 2 加法器和一个向编码器输出的连续转向器，如图 3.4 所示。

编码器每输入一位信息比特将产生三位编码符号。这些编码符号中，第一个输出符号 c_0 是由生成序列 $g_1^{(1)}$ 编码产生的符号，第二个输出符号 c_1 是由生成序列 $g_1^{(2)}$ 编码产生的符号，最后一个输出符号 c_2 是由生成序列 $g_1^{(3)}$ 编码产生的符号。

卷积码编码器的初始状态为全 0。初始状态之后输出的第一个编码符号由生成序列 $g_1^{(1)}$ 编码产生。

例 3.2　图 3.4 所示的卷积码编码器中，三个生成序列分别为：$g_1^{(1)}=(101101111)$，$g_1^{(2)}=(110110011)$，$g_1^{(3)}=(111001001)$。每路接入信道包含 96 比特（20 ms 帧，数据率为4 800 b/s），由 88 位信息比特和 8 位编码尾比特组成，如图 3.3 所示。

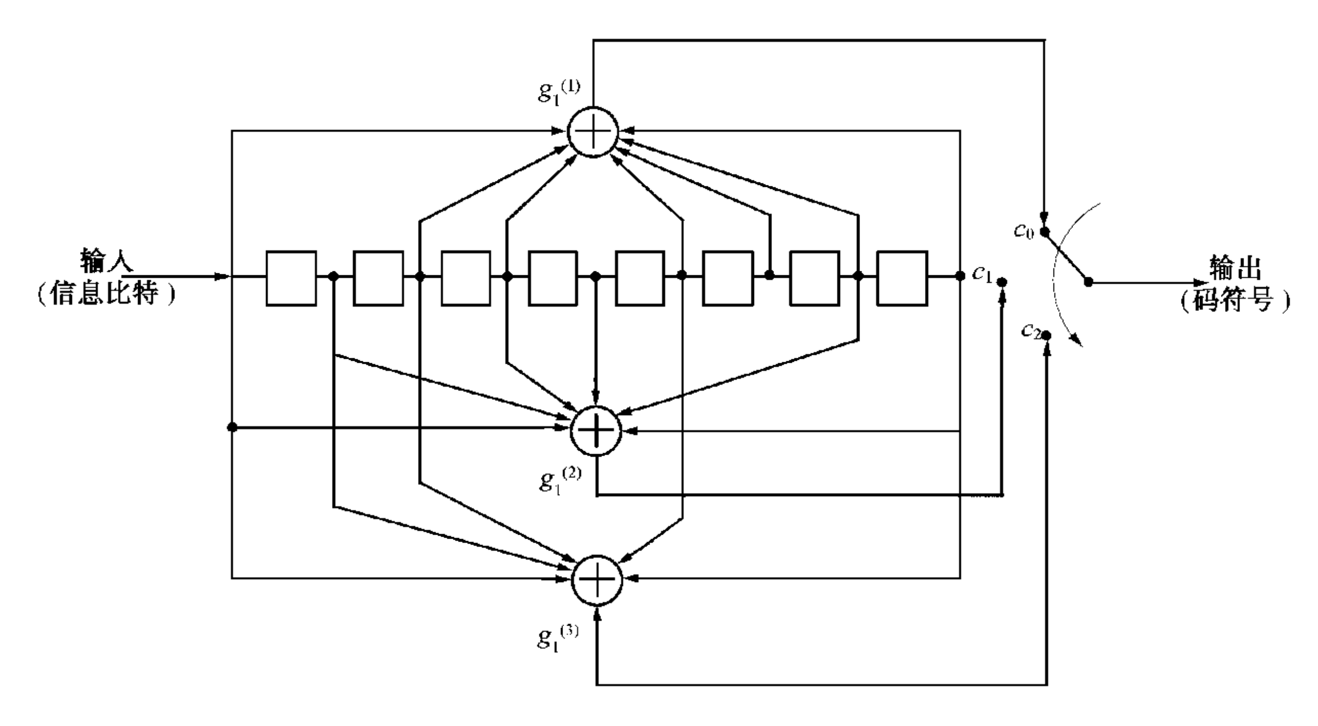
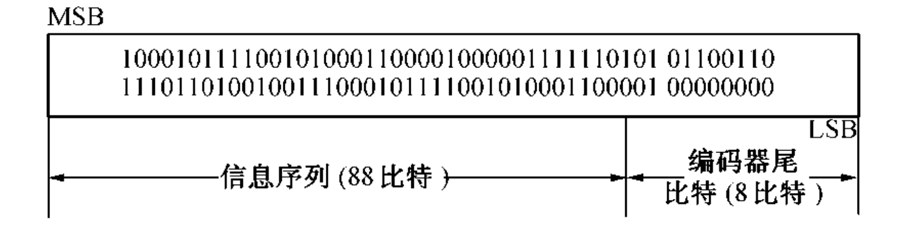


图 3.4　(3,1,8)卷积编码器

假设 88 比特信息序列和 8 位全 0 编码比特如下所示。

帧数据序列（96 比特）



该接入信道帧将输入卷积编码器，输入比特顺序是开始于帧序列的最左边比特的 MSB（最高有效位），截止于最右边位置的 LSB（最低有效位）。

编码器输出符号 c_0，c_1 和 c_2 按照下列等式计算：

$$
\begin{aligned}
c_0 &= d_8+d_6+d_5+d_3+d_2+d_1+d_0 &\quad \text{利用 } g_1^{(1)}=(101101111)\\
c_1 &= d_8+d_7+d_5+d_4+d_1+d_0 &\quad \text{利用 } g_1^{(2)}=(110110011)\\
c_2 &= d_8+d_7+d_6+d_3+d_0 &\quad \text{利用 } g_1^{(3)}=(111001001)
\end{aligned}
\tag{3.1}
$$

第一个输出符号 c_0 是由 $g_1^{(1)}=557$（八进制）编码产生的符号，第二个输出符号 c_1 是由生成序列 $g_1^{(2)}=663$（八进制）编码产生的符号，第三个输出符号 c_2 是由生成序列 $g_1^{(3)}=771$（八进制）编码产生的符号。

利用方程(3.1)，对应于输入 M 的 96 比特帧的编码符号(c_0，c_1，c_2)计算结果如表 3.1 所列。

表 3.1 对应 96 位帧输入 M 的码符号输出 c

帧输入比特 M	卷积编码器内容	输出符号(c_0,c_1,c_2)	帧输入比特 M	卷积编码器内容	输出符号(c_0,c_1,c_2)
1	100000000	111	1	101100110	110
0	010000000	011	1	110110011	001
0	001000000	101	1	111011001	001
0	000100000	110	0	111101100	001
1	100010000	101	1	101110110	100
0	010001000	110	1	110111011	100
1	101000100	110	0	011011101	010
1	110100010	100	1	101101110	011
1	111010001	100	0	010110111	010
1	111101000	010	0	001011011	011
0	011110100	110	1	100101101	111
0	001111010	010	0	010010110	011
1	100111101	101	0	001001011	001
0	010011110	110	1	100100101	010
1	101001111	010	1	110010010	000
0	010100111	000	1	111001001	011
0	001010011	110	0	011100100	100
0	000101001	100	0	001110010	111
1	100010100	001	0	000111001	110
1	110001010	111	1	100011100	100
0	011000101	101	0	010001110	100
0	001100010	101	1	101000111	111
0	000110001	011	1	110100011	011
0	000011000	111	1	111010001	100
1	100001100	110	1	111101000	010
0	010000110	001	0	011110100	110
0	001000011	100	0	101111010	010
0	000100001	001	1	100111101	101
0	000010000	010	0	010011110	110
0	000001000	101	1	101001111	010
1	100000100	011	0	010100111	000
1	110000010	010	0	001010011	110
1	111000001	110	0	000101001	100
1	111100000	111	1	100010100	001
1	111110000	101	1	110001010	111
1	111111000	000	0	011000101	101
0	011111100	011	0	001100010	101
1	101111110	001	0	000110001	011
0	010111111	111	0	000011000	111
1	101011111	000	1	100001100	110
0	010101111	101	0	010000110	001

（续表）

帧输入比特 M	卷积编码器内容	输出符号(c_0,c_1,c_2)	帧输入比特 M	卷积编码器内容	输出符号(c_0,c_1,c_2)
1	101010111	101	0	001000011	100
1	110101011	110	0	000100001	001
0	011010101	111	0	000010000	010
0	001101010	000	0	000001000	101
1	100110101	000	0	000000100	100
1	110011010	101	0	000000010	110
0	011001101	000	0	000000001	111

将来自模 2 加法器的 96 个编码符号输出(c_0,c_1,c_2)连接起来将得到单一输出序列，然后再输入符号重复器。这样，288 比特编码符号输出可以写成：

```
111011101110101110110100100010110010101110010000
11010000111110110101111110001100001010101011010
110111101000011001111000101101110111000000101000
110001001001100100010011010011111011001010000011
100111110100100111011100010110010101110010000110
100001111101101011111110001100001010101100110111
```

3.1.3　编码符号重复

支持接入信道的数据率固定为 4.8 kb/s。卷积编码器输出的编码符号在交织之前将重复。接入信道每个速率为 4 800 b/s 的编码符号被重复一次，即每个符号连续出现两次。重复之后的编码符号将输入 3.1.4 节要讲述的块交织器。

下图显示了连续重复两次后经过交织的编码符号。

```
111111001111110011111100110011111100111100110000
110000001100111100001100110011111100001100000000
111100110000000011111111110011110011001111111111
111100000011110000000011001100110011001111001100
111100111111110011000000001111000011111111000000
110011110011111100111111000000000000110011000000
111100000011000011000011110000110000001100001111
001100001111111111001111000011001100000000001111
110000111111111100110000110000111111001111110000
001100111100001100110011111100001100000000111100
110000000011111111110011110011001111111111111100
000011110000000011001100110011110000111100111111
```

3.1.4　块交织器

使用块交织器的目的不仅在于纠正多径环境下发送数据的突发性错误，还用来获得额外冗余提高稳定性能。移动台对所有接入信道编码符号进行交织，然后进行调制和发送。这里使用持续

时间为 20 ms 帧的块交织器。该交织器是一个 32 行、18 列共 576 个元素的矩阵(参见表 3.2)。

接入信道编码符号将按以下行列顺序从交织器输出:

1 17 9 25 5 21 13 29 3 19 11 27 7 23 15 31

2 18 10 26 6 22 14 30 4 20 12 28 8 24 16 32

表 3.2 用于接入信道或反向业务信道的**4 800 b/s**交织器存储(**TIA/EIA/IS-95**)

1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241	257	273
1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241	257	273
2	18	34	50	66	82	98	114	130	146	162	178	194	210	226	242	258	274
2	18	34	50	66	82	98	114	130	146	162	178	194	210	226	242	258	274
3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243	259	275
3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243	259	275
4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244	260	276
4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244	260	276
5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245	261	277
5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245	261	277
6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246	262	278
6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246	262	278
7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247	263	279
7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247	263	279
8	24	40	56	72	88	104	120	136	152	168	184	200	216	232	248	264	280
8	24	40	56	72	88	104	120	136	152	168	184	200	216	232	248	264	280
9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249	265	281
9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249	265	281
10	26	42	58	74	90	106	122	138	154	170	186	202	218	234	250	266	282
10	26	42	58	74	90	106	122	138	154	170	186	202	218	234	250	266	282
11	27	43	59	75	91	107	123	139	155	171	187	203	219	235	251	267	283
11	27	43	59	75	91	107	123	139	155	171	187	203	219	235	251	267	283
12	28	44	60	76	92	108	124	140	156	172	188	204	220	236	252	268	284
12	28	44	60	76	92	108	124	140	156	172	188	204	220	236	252	268	284
13	29	45	61	77	93	109	125	141	157	173	189	205	221	237	253	269	285
13	29	45	61	77	93	109	125	141	157	173	189	205	221	237	253	269	285
14	30	46	62	78	94	110	126	142	158	174	190	206	222	238	254	270	286
14	30	46	62	78	94	110	126	142	158	174	190	206	222	238	254	270	286
15	31	47	63	79	95	111	127	143	159	175	191	207	223	239	255	271	287
15	31	47	63	79	95	111	127	143	159	175	191	207	223	239	255	271	287
16	32	48	64	80	96	112	128	144	160	176	192	208	224	240	256	272	288
16	32	48	64	80	96	112	128	144	160	176	192	208	224	240	256	272	288

如表 3.2 所示,编码符号必须按列完全填满 32 × 18 的矩阵来写入块交织器。为了方便,我们

按如下方式排列交织器的输入。

符号编号	块交织器输入			
01 ~ 32	11101110	11101011	10110100	10001011
33 ~ 64	00101011	10010000	11010000	11111011
65 ~ 96	01011111	11000110	00010101	01011010
97 ~ 128	11011110	10000110	01111000	10110111
129 ~ 160	01110000	00101000	11000100	10011001
161 ~ 192	00010011	01001111	10110010	10000011
193 ~ 224	10011111	01001001	11011100	01011001
225 ~ 256	01011100	10000110	10000111	11011010

由行创建的块交织器输出可以使用例 3.3 所示的交织器输出符号计算。

例 3.3　利用交织器行数[λ]，可以如下计算输出符号：

$\lambda=1$,	1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241	257	273
	1	1	0	1	0	0	1	0	0	1	0	1	1	1	0	1	1	1
$\lambda=17$,	9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249	265	281
	1	1	1	1	1	0	1	1	0	1	0	1	0	0	1	1	0	0
$\lambda=9$,	5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245	261	277
	1	0	1	0	1	0	1	1	0	0	0	0	1	1	1	0	1	1

这样，对应于行数 λ 的块交织器输出可按如下所示方式计算：

λ	交织器输出符号	λ	交织器输出符号
[1]	1 1 0 1 0 0 1 0 0 1 0 1 1 1 0 1 1 1	[2]	1 1 0 1 0 0 1 0 0 1 0 1 1 1 0 1 1 1
[17]	1 1 1 1 1 0 1 1 0 1 0 1 0 0 1 1 0 0	[18]	1 1 1 1 1 0 1 1 0 1 0 1 0 0 1 1 0 0
[9]	1 0 1 0 1 0 1 1 0 0 0 0 1 1 1 0 1 1	[10]	1 0 1 0 1 0 1 1 0 0 0 0 1 1 1 0 1 1
[25]	1 1 0 1 0 1 0 0 1 1 1 0 1 1 0 1 0 0	[26]	1 1 0 1 0 1 0 0 1 1 1 0 1 1 0 1 0 0
[5]	1 1 1 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1	[6]	1 1 1 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1
[21]	1 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 1 1	[22]	1 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 1 1
[13]	1 0 1 0 1 0 1 0 0 0 1 1 1 0 0 1 1 1	[14]	1 0 1 0 1 0 1 0 0 0 1 1 1 0 0 1 1 1
[29]	1 1 0 1 1 1 1 1 0 0 1 1 0 0 1 1 0 1	[30]	1 1 0 1 1 1 1 1 0 0 1 1 0 0 1 1 0 1
[3]	1 0 0 1 1 0 1 1 1 1 0 0 0 1 1 0 1 0	[4]	1 0 0 1 1 0 1 1 1 1 0 0 0 1 1 0 1 0
[19]	1 0 0 1 1 1 0 0 0 0 1 0 1 1 0 1 0 0	[20]	1 0 0 1 1 1 0 0 0 0 1 0 1 1 0 1 0 0
[11]	1 1 0 0 1 1 1 0 0 1 0 0 1 1 1 1 1 0	[12]	1 1 0 0 1 1 1 0 0 1 0 0 1 1 1 1 1 0
[27]	0 0 0 0 1 0 1 1 0 0 1 0 0 0 1 0 0 1	[28]	0 0 0 0 1 0 1 1 0 0 1 0 0 0 1 0 0 1
[7]	0 1 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1 0	[8]	0 1 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1 0
[23]	0 0 1 1 0 1 0 1 0 1 0 0 0 1 0 1 1 1	[24]	0 0 1 1 0 1 0 1 0 1 0 0 0 1 0 1 1 1
[15]	0 0 1 0 1 1 0 0 0 0 1 0 1 0 0 1 0 1	[16]	0 0 1 0 1 1 0 0 0 0 1 0 1 0 0 1 0 1
[31]	1 1 0 1 0 0 0 1 0 1 1 1 1 1 0 0 0 1	[32]	1 1 0 1 0 0 0 1 0 1 1 1 1 1 0 0 0 1

最后可以连接对应于 λ 的所有行输出而得到交织器输出序列，它们是：

这个二进制序列表示交织器输出，作为正交调制的输入。正如前面提到的，编码符号重复一

110100100101110111111110110101001100101010110000

111011110101001110110100111000011001000011100100

011000000011101010100011100111110111110011001101

100110111100011010100111000010110100110011100100

111110000010110010001001010111111011111010001101

010100010111001011000010100101110100010111110001

110100100101110111111110110101001100101010110000

111011110101001110110100111000011001000011100101

011000000011101010100011100111110111110011001101

100110111100011010100111000010110100110011100100

111110000010110010001001010111111011111010001101

010100010111001011000010100101110100010111110001

次，然后在接入信道发送，如表 3.2 所示。

这些重复的编码符号按照图 3.5 所示的发送结构传输。

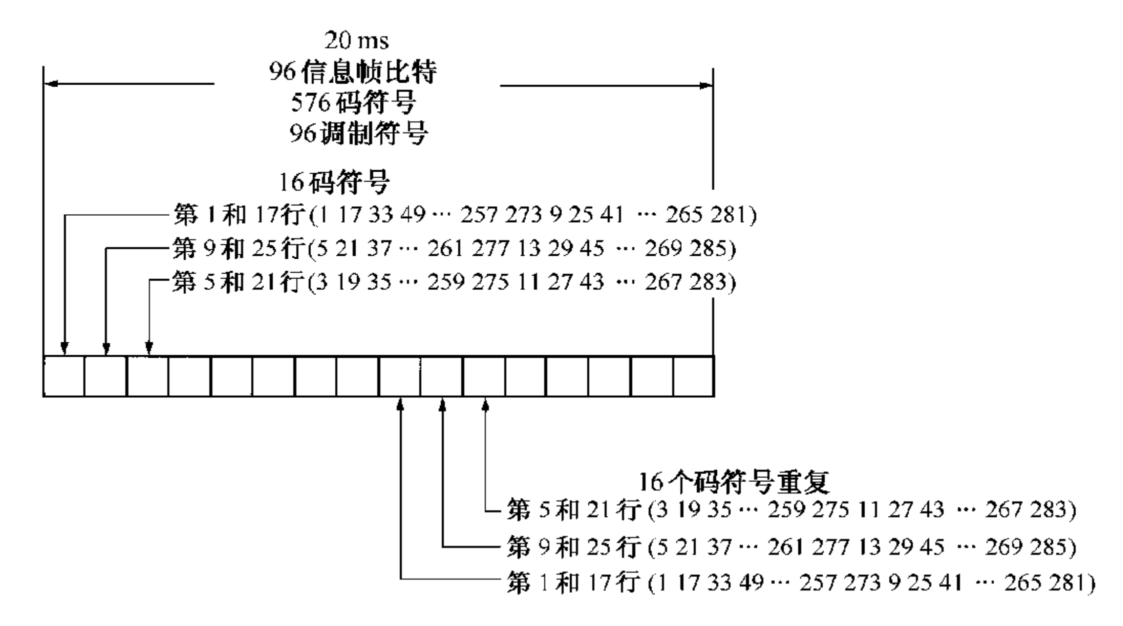


图 3.5 接入信道发送结构（利用表 3.2 和交织器行顺序）

3.1.5 64 阶正交调制

接入信道使用 64 阶正交调制。调制符号是利用 Walsh 函数产生的 64 个相互正交的波形之一。这些调制符号由表 2.4 给出。

每 6 个编码符号对应传输 64 种可能的调制符号之一。观察图 3.6，由于块交织器输出的编码符号速率为 28.8 ks/s，正交调制器符号输出速率变成 28.8/6 = 4.8 ks/s 或 4.8 × 64 = 307.2 kc/s。

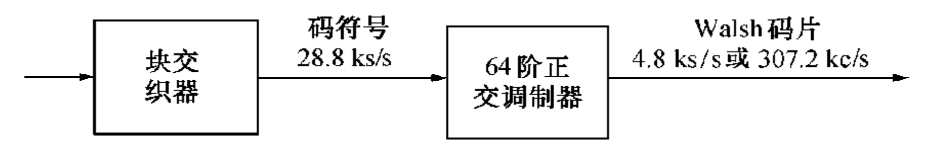


图 3.6 正交调制

对特定 MSI（调制符号指数），根据下列公式选择调制符号：

$$\mathrm{MSI} = c_0 + 2c_1 + 4c_2 + 8c_3 + 16c_4 + 32c_5$$

其中，c_i，$0 \leqslant i \leqslant 5$，表示二进制数值编码符号，用来生成调制符号指数。

传输单个调制符号需要的时间间隔为 1/4800 = 208.333 μs。Walsh 码片定义为相关调制符号的 1/64 时间间隔，即 Walsh 码片间隔为 208.333/64 = 3.2552 μs。

例 3.4 由于每 6 个编码符号对应传输 64 种可能的调制符号之一，最好将 64 阶正交调制器的输入分成 6 个符号的等长度，如下图所示。

正交调制器输入（576 符号）

```
110100 100101 110111 111110 110101 001100 101010 110000
111011 110101 001110 110100 111000 011001 000011 100100
011000 000011 101010 100011 100111 110111 110011 001101
100110 111100 011010 100111 000010 110100 110011 100100
111110 000010 110010 001001 010111 111011 111010 001101
010100 010111 001011 000010 100101 110100 010111 110001
110100 100101 110111 111110 110101 001100 101010 110000
111011 110101 001110 110100 111000 011001 000011 100100
011000 000011 101010 100011 100111 110111 110011 001101
100110 111100 011010 100111 000010 110100 110011 100100
111110 000010 110010 001001 010111 111011 111010 001101
010100 010111 001011 000010 100101 110100 010111 110001
```

下面看看如何计算正交调制器输出（每 6 144 个符号输入对应 512 个符号输出）。

利用公式 $MSI = \sum_{i=0}^{5} 2^i c_i$ 对应正交输入（576 个符号）最上面行（48 符号）的正交调制器输出（512 个符号）来进行一个简单的演示。

6 符号输入	MSI	64 阶输出
110100	11	0110011010011001011001101001100101100110100110010110011010011001
100101	41	0101010110101010010101011010101010101010010101011010101001010101
110111	59	0110011010011001100110010110011010011001011001100110011010011001
111110	31	0110100110010110100101100110100101101001100101101001011001101001
110101	43	0110011010011001011001101001100110011001011001101001100101100110
001100	12	0000111111110000000011111111000000001111111100000000111111110000
101010	21	0101101001011010101001011010010101011010010110101010010110100101
110000	3	0110011001100110011001100110011001100110011001100110011001100110

因此，6 114 个符号正交调制器部分输出的 512 个符号为下列表中所示：

```
0110011010011001011001101001100101100110100110010110011010011001
0101010110101010010101011010101010101010010101011010101001010101
0110011010011001100110010110011010011001011001100110011010011001
0110100110010110100101100110100101101001100101101001011001101001
0110011010011001011001101001100110011001011001101001100101100110
0000111111110000000011111111000000001111111100000000111111110000
0101101001011010101001011010010101011010010110101010010110100101
0110011001100110011001100110011001100110011001100110011001100110
```

3.1.6 长码直接序列扩展

接入信道发送数据时，采用长码直接序列扩展，然后再发送。扩展操作包括 64 阶正交调制器输出序列和长 PN 码（参见图 3.7）的模 2 加。长码周期为 $2^{42}-1$ 个码片，由码生成式的LFSR抽头多项式 $p(x)$ 指定。

$$p(x)=1+x+x^2+x^3+x^5+x^6+x^7+x^{10}+x^{16}+x^{17}+x^{18}+x^{19}+x^{21}+x^{22}+x^{25}+x^{26}+x^{27}+x^{31}+x^{33}+x^{35}+x^{42} \quad (3.2)$$

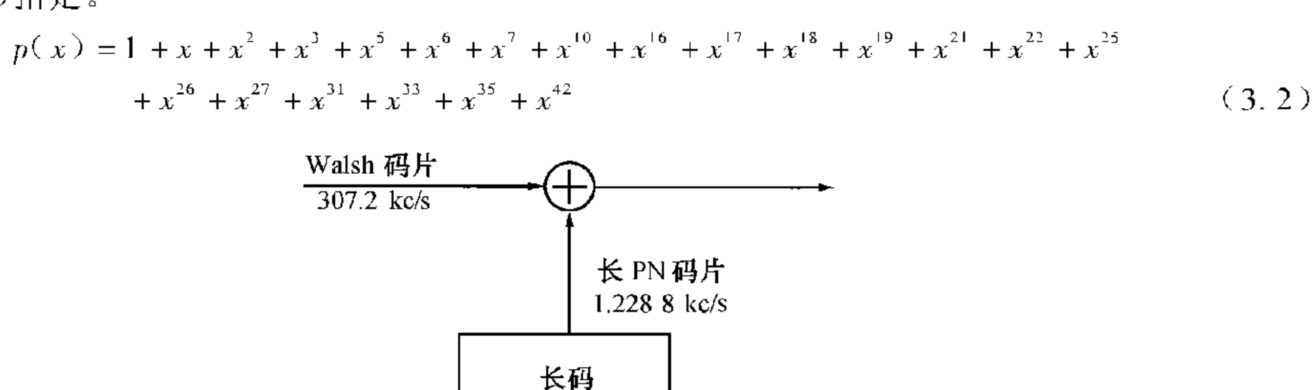


图 3.7 64 阶正交调制器输出和长码模 2 加的直接序列扩展

每一个长码 PN 码片由图 3.8 所示的 42 位掩码和序列生成式的 42 级 LFSR“与”结果的模 2 和产生。长码根据移动台传输的不同信道类型而变化。接入信道长码掩码如图 3.9 所示。接入信道发送时，掩码格式设置如下：

$M_{41}\sim M_{33}$：设置成 110001111

$M_{32}\sim M_{28}$：设置成选定的接入信道编号

$M_{27}\sim M_{25}$：为关联寻呼信道设置码道编号

$M_{24}\sim M_{9}$：为当前基站设置 BASE-ID 值

$M_{8}-M_{0}$：为当前 CDMA 信道设置 PILOT-PN 值

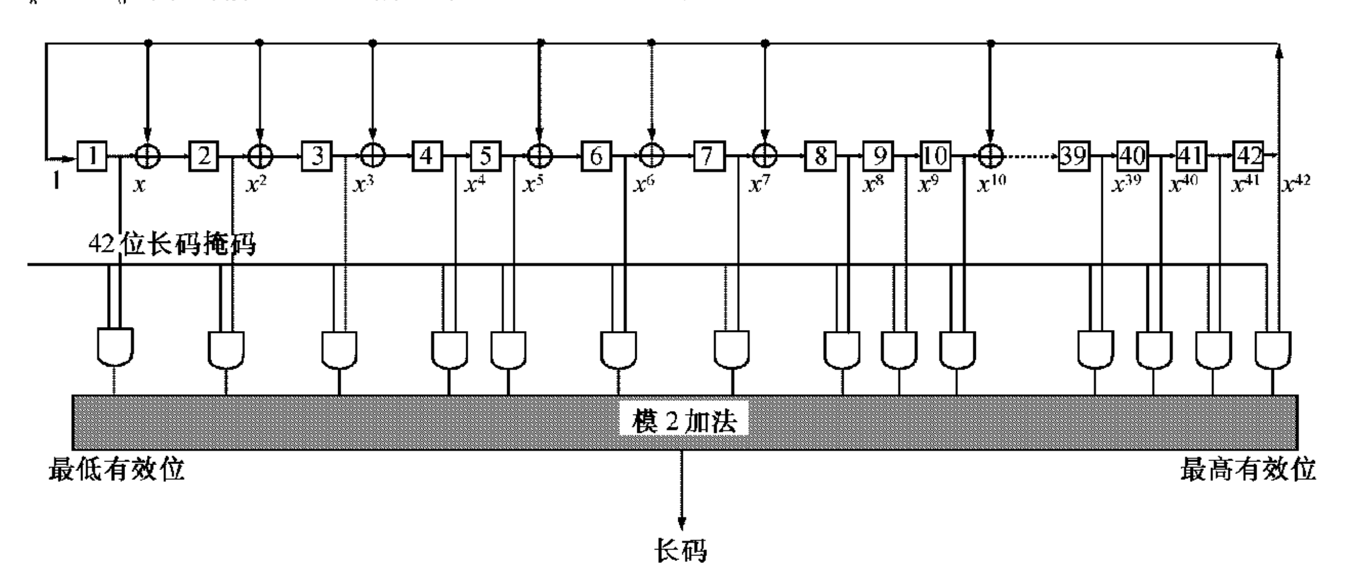
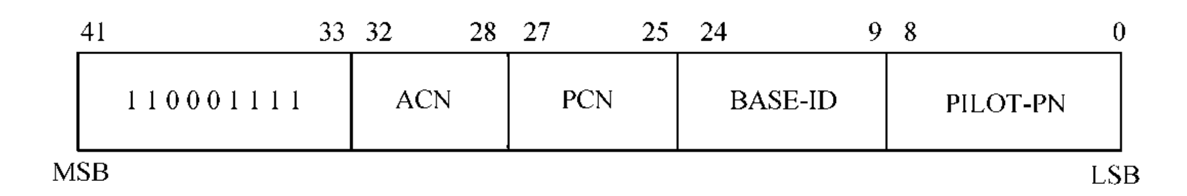


图 3.8 长码发生器（TIA/EIA/IS-95 之后）



ACN:**接入信道编号**
PCN:**寻呼信道编号**
BASE-ID:**基站标识**
PILOT-PN:**前向 CDMA 信道 PN 偏移**

图 3.9　接入信道长码掩码

下面适当探讨一下正交调制器输出的 Walsh 码片和长码发生器产生的长 PN 码片的“异或”运算，如图 3.7 所示。

每一个 PN 码片由 42 位掩码和 42 级发生器的模 2 和产生。由长码掩码激活的长码发生器产生 1.228 8 kc/s 的 PN 码片，如图 3.8 所示。

例 3.5　长码掩码头设置为（110001111），同时假设 ACN 设置为（00010），PCN 设置为（011），BASE-ID 设置为（0100011010011101），PILOT-PN 设置为（100111001）。则 42 位长码掩码由最低有效位（LSB）到最高有效位（MSB）为（100111001101110010110001011001000111100011）。

参照方程 3.2，42 位发生器抽头序列可以表示为（111101110010000011110110011100010101000000）。

假设长码发生器初始内容设置为（100000000000000000000000000000000000000000）。长码掩码的每个比特和发生器每级输出比特都应经过“与”门电路。“与”门输出必须由模 2 加法器经过“异或”运算。长 PN 序列可采用表 3.3 所示的步骤产生。

表 3.3　长 PN 码产生步骤

移位编号	生成器内容	“与”操作后 1 的个数	输出
0	100000000000000000000000000000000000000000	1	1
1	010000000000000000000000000000000000000000	0	0
2	001000000000000000000000000000000000000000	0	0
3	000100000000000000000000000000000000000000	1	1
4	000010000000000000000000000000000000000000	1	1
5	000001000000000000000000000000000000000000	1	1
6	000000100000000000000000000000000000000000	0	0
7	000000010000000000000000000000000000000000	0	0
8	000000001000000000000000000000000000000000	1	1
9	000000000100000000000000000000000000000000	1	1
10	000000000010000000000000000000000000000000	0	0
11	000000000001000000000000000000000000000000	1	1
12	000000000000100000000000000000000000000000	1	1
13	000000000000010000000000000000000000000000	1	1
14	000000000000001000000000000000000000000000	0	0
15	000000000000000100000000000000000000000000	0	0
16	000000000000000010000000000000000000000000	1	1

（续表）

移位编号	生成器内容	“与”操作后 1 的个数	输出
17	000000000000000001000000000000000000000000	0	0
18	000000000000000000100000000000000000000000	1	1
19	000000000000000000010000000000000000000000	1	1
20	000000000000000000001000000000000000000000	0	0
21	000000000000000000000100000000000000000000	0	0
22	000000000000000000000010000000000000000000	0	0
23	000000000000000000000001000000000000000000	1	1
24	000000000000000000000000100000000000000000	0	0
25	000000000000000000000000010000000000000000	1	1
26	000000000000000000000000001000000000000000	1	1
27	000000000000000000000000000100000000000000	0	0
28	000000000000000000000000000010000000000000	0	0
29	000000000000000000000000000001000000000000	1	1
30	000000000000000000000000000000100000000000	0	0
31	000000000000000000000000000000010000000000	0	0
32	000000000000000000000000000000001000000000	0	0
33	000000000000000000000000000000000100000000	1	1
34	000000000000000000000000000000000010000000	1	1
35	000000000000000000000000000000000001000000	1	1
36	000000000000000000000000000000000000100000	1	1
37	000000000000000000000000000000000000010000	0	0
38	000000000000000000000000000000000000001000	0	0
39	000000000000000000000000000000000000000100	0	0
40	000000000000000000000000000000000000000010	1	1
41	000000000000000000000000000000000000000001	1	1
42	111101110010000011110110011100010101000000	10	0
43	011110111001000001111011001110001010100000	10	0
44	001111011100100000111101100111000101010000	12	0
45	000111101110010000011110110011100010101000	11	1
46	000011110111001000001111011001110001010100	9	1
47	000001111011100100000111101100111000101010	8	0
48	000000111101110010000011110110011100010101	10	0
49	111101101100111010110111100111011011001010	15	1
50	011110110110011101011011110011101101100101	12	0
51	110010101001001101011011100101100011110010	11	1
52	011001010100100110101101110010110001111001	10	0
53	110001011000010000100000100101001101111100	9	1
54	011000101100001000010000010010100110111110	8	0
55	001100010110000100001000001001010011011111	8	0
56	111011111001000001110010011000111100101111	13	1
57	100000001110100011001111010000001011010111	11	1

（续表）

移位编号	生成器内容	“与”操作后 1 的个数	输出
58	101101110101010010010001110100010000101011	13	1
59	101011001000101010111110100110011101010101	11	1
60	101000010110010110101001001111011011101010	12	0
61	010100001011001011010100100111101101110101	10	0
62	110111110111100110011100001111100011111010	15	1
63	011011111011110011001110000111110001111101	11	1
64	110000000111111101001000101111110110111110	16	0
65	011000000011111110100100010111111011011111	11	1
66	110001110001111101010010001011101110011111	12	0
67	100101001010111101011111011001100010001111	14	0
68	101111010111011101011001110000100100000111	13	1
69	101010011001101101011010100100000111000011	11	1
70	101000111110110101011011001110010110100001	12	0
71	101001101101011001011011111011011110010000	13	1
72	010100110110101100101101111101101111001000	11	1
73	001010011011010110010110111110110111100100	12	0
74	000101001101101011001011011111011011110010	15	1
75	000010100110110101100101101111101101111001	12	0
76	111100100001011001000100101011100011111100	9	1
77	011110010000101100100010010101110001111110	9	1
78	001111001000010110010001001010111000111111	12	0
79	111010010110001000111110111001001001011111	11	1
80	100000111001000111101001000000110001101111	10	0
81	101101101110100000000010111100001101110111	13	1
82	101011000101010011110111000010010011111011	15	1
83	101000010000101010001101111101011100111101	10	0
84	101001111010010110110000100010111011011110	10	0
85	010100111101001011011000010001011101101111	13	1
86	110111101100100110011010010100111011110111	15	1
87	100110000100010000111011010110001000111011	12	0
88	101110110000000101110101111011101000101110 1	10	0
89	101010101010000110000011100111111101101110	10	0
90	010101010101000011000001110011111110110111	13	1
91	110111011000100010010110100101101010011011	12	0
92	100110011110010010111101001110100000001101	12	0
93	101110111101001010101000111011000101000110	14	0
94	010111011110100101010100011101100010100011	14	0
95	110110011101010001011100010010100100010001	11	1
96	100110111100101011011000010101000111001000	13	1
97	010011011110010101101100001010100011100100	10	0
98	001001101111001010110110000101010001110010	11	1

（续表）

移位编号	生成器内容	“与”操作后 1 的个数	输出
99	000100110111100101011011000010101000111001	8	0
.	.	.	.
.	.	.	.
.	.	.	.

这样，长码序列可以计算为

```
10011100110111001011000101100100011110001100011001010l
001111001101001101101011010110011000100001101 0……
```

由于 PN 码片速率为 1.228 8 Mc/s，Walsh 码片速率为 307.2 kc/s，可以得到比率为 PN 码片/Walsh 码片 = $1.2288 \times 10^6 / 307.2 \times 10^3 = 4$。

因此，为了进行正交调制器输出（O.M）和长码输出（L.C）的“异或”运算，将长码序列划分成 4 位一组等长的段是很方便的。

O.M	0	1	1	0	0	1	1	0	1	0	0	1	…
L.C	1001	1100	1101	1100	1011	0001	0110	0100	0111	1000	1100	0110	…
EX-OR（⊕）	1001	0011	0010	1100	1011	1110	1001	0100	1000	1000	1100	1001	…

这样，直接序列扩展模 2 加法器输出序列为

```
10010011 00101100 10111110 10010100 10001000 11001001……
```

3.1.7 正交相位扩展

直接序列扩展之后，接入信道进行正交相位扩展，如图 3.10 所示。进行扩展的序列是 0 偏移的 I 和 Q 导频 PN 序列。这些 PN 序列周期为 2^{15} 个码片，分别基于下列 LFSR 抽头多项式。

对同相序列，多项式为

$$P_I(x) = 1 + x^5 + x^7 + x^8 + x^9 + x^{13} + x^{15} \tag{3.3}$$

对正交相序列，多项式为

$$P_Q(x) = 1 + x^3 + x^4 + x^5 + x^6 + x^{10} + x^{11} + x^{12} + x^{15} \tag{3.4}$$

15 级 LFSR 基于方程 3.3 和 3.4，分别如图 3.11 和 3.12 所示。

例 3.6 计算 I 和 Q 导频 PN 序列扩展数据

数据序列 I 和 Q 是 I 和 Q 导频 PN 序列和长码直接扩展生成序列进行“异或”运算产生的序列。

首先考虑导频 PN 码产生。从方程 3.3 和 3.4 可以得到，抽头序列分别为 P_I = (100001011100010)，P_Q = (100011110001110 0)。

假设发生器初始内容为（100000000000000），LFSR 产生的 $\overline{P}_I$ 和 $\overline{P}_Q$ 移位序列如表 3.4 所示。

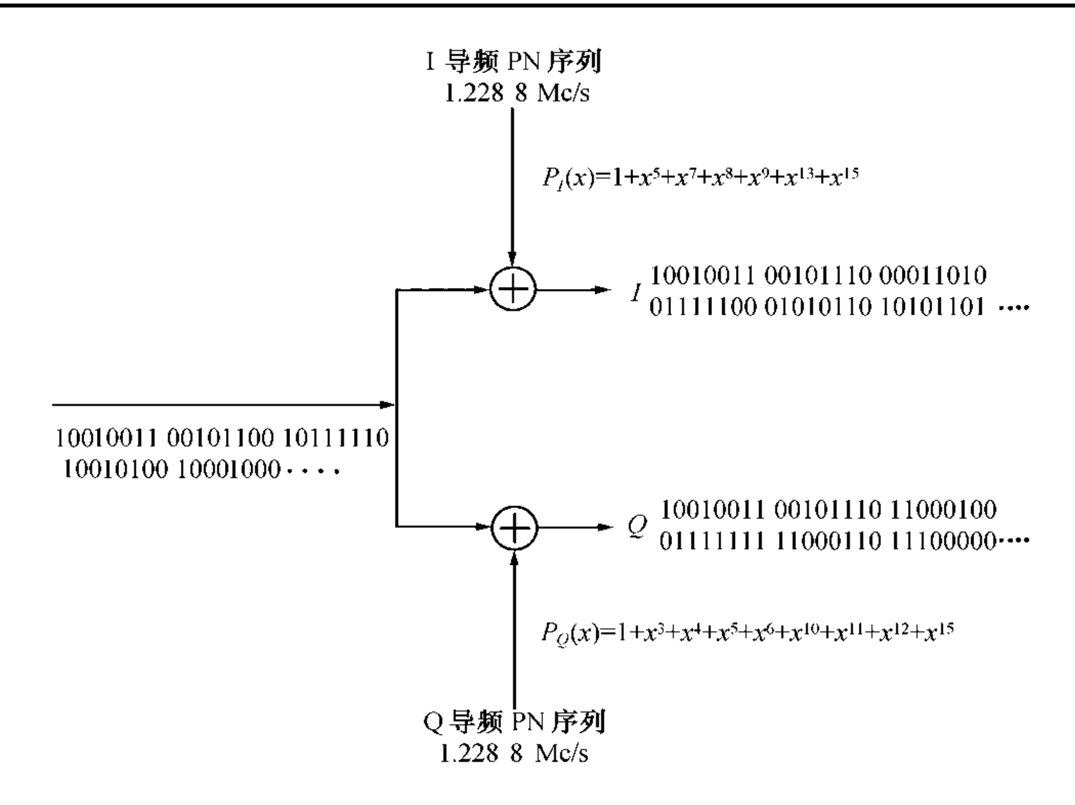


图 3.10　四相正交扩展

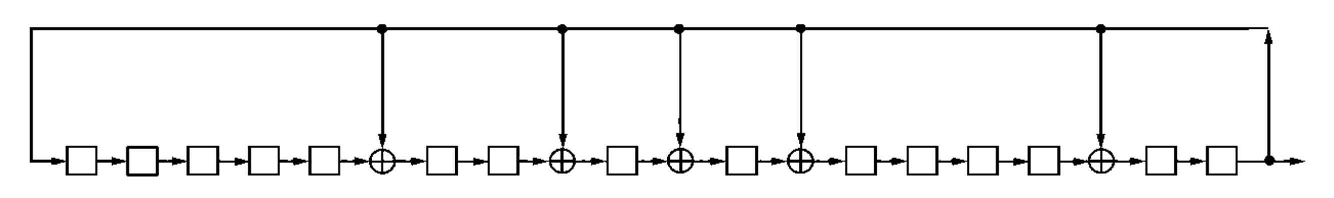


图 3.11　基于抽头多项式 $P_I(x)$ 的 15 级 LFSR

$$P_I(x) = 1 + x^5 + x^7 + x^8 + x^9 + x^{13} + x^{15}$$

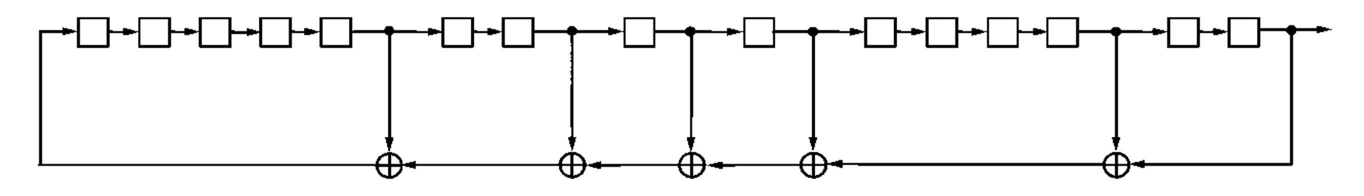


图 3.12　基于互补抽头多项式 $i(x)$ 的 15 级 LFSR

$$i(x) = 1 + x^2 + x^6 + x^7 + x^8 + x^{10} + x^{15}$$

表 3.4　四相正交导频 PN 序列 $\bar{P}_I$ 和 $\bar{P}_Q$ 的产生

移位编号	LFSR 内容	P_I	移位编号	LFSR 内容	P_Q
0	100000000000000		0	100000000000000	
1	010000000000000	0	1	010000000000000	0
2	001000000000000	0	2	001000000000000	0
3	000100000000000	0	3	000100000000000	0
4	000010000000000	0	4	000010000000000	0
5	000001000000000	0	5	000001000000000	0
6	000000100000000	0	6	000000100000000	0
7	000000010000000	0	7	000000010000000	0
8	000000001000000	0	8	000000001000000	0

（续表）

移位编号	LFSR 内容	P_I	移位编号	LFSR 内容	P_Q
9	000000000100000	0	9	000000000100000	0
10	000000000010000	0	10	000000000010000	0
11	000000000001000	0	11	000000000001000	0
12	000000000000100	0	12	000000000000100	0
13	000000000000010	0	13	000000000000010	0
14	000000000000001	0	14	000000000000001	0
15	100001011100010	1	15	100111100011100	1
16	010000101110001	0	16	010011110001110	0
17	101001001011010	1	17	001001111000111	0
18	010100100101101	0	18	100011011111111	1
19	101011001110100	1	19	110110001100011	1
20	010101100111010	0	20	111100100101101	1
21	001010110011101	0	21	111001110001010	1
22	100100000101100	1	22	011100111000101	0
23	010010000010110	0	23	101001111111110	1
24	001001000001011	0	24	010100111111111	0
25	100101111100111	1	25	101101111100011	1
26	110011100010001	1	26	110001011101101	1
27	111000101101010	1	27	111111001101011	1
28	011100010110101	0	28	011111100110101	0
29	101111010111000	1	29	101000010000110	1
30	010111101011100	0	30	010100001000011	0
31	001011110101110	0	31	101101100111101	1
32	000101111010111	0	32	110001010000010	1
33	100011100001001	1	33	011000101000001	0
34	110000101100110	1	34	101011110111100	1
35	011000010110011	0	35	010101111011110	0
36	101101010111011	1	36	001010111101111	0
37	110111110111111	1	37	100010111101011	1
38	111010100111101	1	38	110110111101001	1
39	111100001111100	1	39	111100111101000	1
40	011110000111110	0	40	011110011110100	0
41	001111000011111	0	41	001111001111010	0
42	100110111101101	1	42	000111100111101	0
43	110010000010100	1	43	100100010000010	1
44	011001000001010	0	44	010010001000001	0
45	001100100000101	0	45	101110100111100	1
46	100111001100000	1	46	010111010011110	0
47	010011100110000	0	47	001011101001111	0
48	001001110011000	0	48	100010010111011	1
49	000100111001100	0	49	110110101000001	1

（续表）

移位编号	LFSR 内容	P_I	移位编号	LFSR 内容	P_Q
50	000010011100110	0	50	111100110111100	1
51	000001001110011	0	51	011110011011110	0
52	100001111011011	1	52	001111001101111	0
53	110001100001111	1	53	100000000101011	1
54	111001101100101	1	54	110111100001001	1
55	111101101010000	1	55	111100010011000	1
56	011110110101000	0	56	011110001001100	0
57	001111011010100	0	57	001111000100110	0
58	000111101101010	0	58	000111100010011	0
59	000011110110101	0	59	100100010010101	1
60	100000100111000	1	60	110101101010110	1
61	010000010011100	0	61	011010110101011	1
62	001000001001110	0	62	101010111001001	1
63	000100000100111	0	63	110010111111000	1
64	100011011110001	1	64	011001011111100	0
65	110000110011010	1	65	001100101111110	0
66	011000011001101	0	66	000110010111111	0
67	101101010000100	1	67	100100101000011	1
68	010110101000010	0	68	110101110111101	1
69	001011010100001	0	69	111101011000010	1
70	100100110110010	1	70	011110101100001	0
71	010010011011001	0	71	101000110101100	1
72	101000010001110	1	72	010100011010110	0
73	010100001000111	0	73	001010001101011	0
74	101011011000001	1	74	100010100101001	1
75	110100110000010	1	75	110110110001000	1
76	011010011000001	0	76	011011011000100	0
77	101100010000010	1	77	001101101100010	0
78	010110001000001	0	78	000110110110001	0
79	101010011000010	1	79	100100111000100	1
80	010101001100001	0	80	010010011100010	0
81	101011111010010	1	81	001001001110001	0
82	010101111101001	0	82	100011000100100	1
83	101011100010110	1	83	010001100010010	0
84	010101110001011	0	84	001000110001001	0
85	101011100100111	1	85	100011111011000	1
86	110100101110001	1	86	010001111101100	0
87	111011001011010	1	87	001000111110110	0
88	011101100101101	0	88	000100011111011	0
89	101111101110100	1	89	100101101100001	1
90	010111110111010	0	90	110101010101100	1

（续表）

移位编号	LFSR 内容	P_I	移位编号	LFSR 内容	P_Q
91	001011111011101	0	91	011010101010110	0
92	100100100001100	1	92	001101010101011	0
93	010010010000110	0	93	100001001001001	1
94	001001001000011	0	94	110111000111000	1
95	100101111000011	1	95	011011100011100	0
96	110011100000011	1	96	001101110001110	0
97	111000101100011	1	97	000110111000111	0
98	111101001010011	1	98	100100111111111	1
99	111111111001011	1	99	110101111100011	1
.				.	
.				.	
.				.	

为了获得 I 和 Q 导频 PN 序列，在 $\bar{P}_I$ 和 $\bar{P}_Q$ 连续输出 14 个 0 之后再插入 1 个 0（↓）。出于对齐的目的，$\bar{P}_I$ 和 $\bar{P}_Q$ 必须在连续 15 个 0 之后插入 1。

■　■↓

$\bar{P}_I$:00000000000000010101001001110100011011110011001000
00111100001000110100101011010101010111010010011111

↓

$\bar{P}_Q$:00000000000000010011110101110101101001110001010011
10011100011011000111010011000100100100011001100011

最后，对直接扩展序列和导频 PN 序列进行"异或"运算，P_I 和 P_Q，正交序列 I 和 Q 可计算如下：

DSS　10010011 00101100 10111110 10010100 10001000 11001001…

⊕

$\bar{P}_I$　00000000 00000001 01010010 01110100 01101111 00110010

I	10010011 00101101 11101100 11100000 11100111 11111011…

DSS　10010011 00101100 10111110 10010100 10001000 11001001…

⊕

$\bar{P}_Q$　00000000 00000001 00111101 01110101 10100111 00010100

Q	10010011 00101101 10000011 11100001 00101111 11011101…

由 Q 导频 PN 序列扩展的数据脉冲 Q 相对于由 I 导频 PN 序列扩展的数据脉冲 I 延时半个 PN 码片时间，即 $1/2(1.2288 \times 10^6) = 406.9$ ns。

这些 I 和 Q 脉冲输入 I 和 Q 基带滤波器，如图 3.2 所示。基带滤波之后，二进制 I 和 Q 数据映

射到 3.1.9 节讨论的相位转移上。

零偏移 I 和 Q 导频 PN 序列周期为 2^{15} 个码片，基于方程 3.3 和 3.4 的 LSFR 抽头多项式$P_I(x)$和 $P_Q(x)$。

相同的 I 和 Q 导频 PN 序列也是分别由互逆多项式产生：$i(x) = x^n P_I(x^{-1})$ 和 $q(x) = x^n P_Q(x^{-1})$。

例 3.7　考虑最大长度为$(2^n - 1, n)$，最小距离 $d_{\min} = 2^n - 1$ 的码。

该码是长度为 2^{n-1} 的汉明码的对偶，通过次数为 n 的本原多项式 $P(x)$ 作为生成多项式得到。最大长度为$(2^n - 1, n)$对偶码由次数为 n 的本原互逆多项式 $i(x) = x^n P(x^{-1})$得到。

例 3.8　图 3.11 和 3.12 分别显示了同相序列基于抽头多项式 $P_I(x)$的 LFSR 和正交相位序列的 $P_Q(x)$。

互逆多项式 $i(x) = x^n P_I(x^{-1})$和 $q(x) = x^n P_Q(x^{-1})$计算如下：

$$i(x) = x^{15}(1 + x^{-5} + x^{-7} + x^{-8} + x^{-9} + x^{-13} + x^{-15})$$

$$= 1 + x^2 + x^6 + x^7 + x^8 + x^{10} + x^{15} \tag{3.5}$$

$$q(x) = x^{15}(1 + x^{-3} + x^{-4} + x^{-5} + x^{-6} + x^{-10} + x^{-11} + x^{-12} + x^{-15})$$

$$= 1 + x^3 + x^4 + x^5 + x^9 + x^{10} + x^{11} + x^{12} + x^{15} \tag{3.6}$$

利用互逆抽头多项式，可以描绘基于公式 3.5 和 3.6 的 LFSR，如图 3.13 和 3.14 所示。

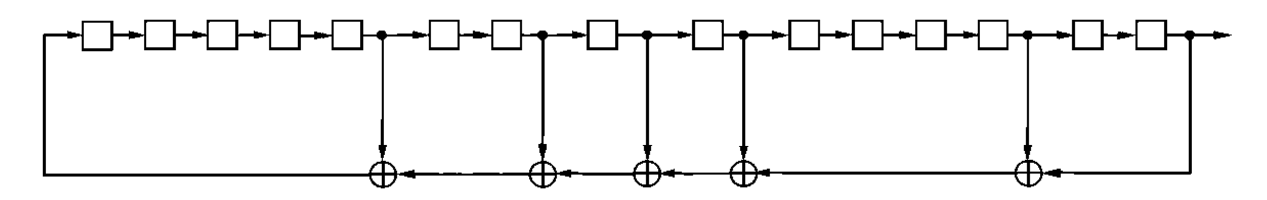


图 3.13　基于互补抽头多项式 $i(x)$的 15 级 LFSR
$i(x) = 1 + x^2 + x^6 + x^7 + x^8 + x^{10} + x^{15}$

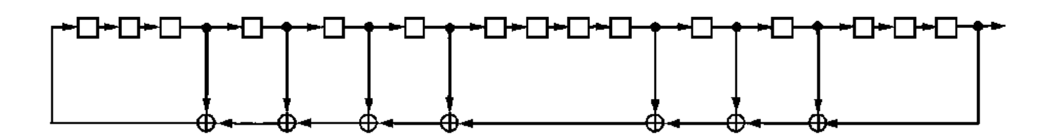


图 3.14　基于互补抽头多项式 $q(x)$的 15 级 LFSR
$q(x) = 1 + x^3 + x^4 + x^5 + x^9 + x^{10} + x^{11} + x^{12} + x^{15}$

例 3.9　假设 $i(n)$和 $q(n)$分别表示基于抽头多项式 $P_I(x)$和 $P_Q(x)$的 LFSR 序列。

这些 $i(n)$和 $q(n)$序列称为周期为 $2^{15} - 1$ 的 LFSR 序列，可利用下面的线性递归操作产生：

$$i(n) = i(n-15) \oplus i(n-10) \oplus i(n-8) \oplus i(n-7) \oplus i(n-6) \oplus i(n-2) \tag{3.7}$$

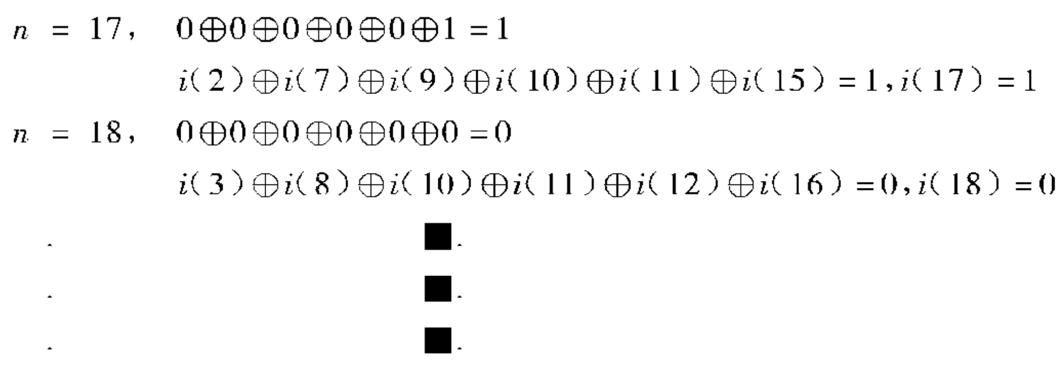
$$q(n) = q(n-15) \oplus q(n-12) \oplus q(n-11) \oplus q(n-10) \oplus q(n-9) - q(n-5)$$
$$\oplus q(n-4) \oplus q(n-3) \tag{3.8}$$

其中 $i(n)$和 $q(n)$表示 0 或 1，$i \leqslant n \leqslant 32\,767$，$i(15) = 1$，$q(15) = 1$，$\oplus$表示模 2 加。

由于 $i(n) = 0, 1 \leqslant n \leqslant 14$ 且 $i(15) = 1$，当 $16 \leqslant n \leqslant 32\,767$ 时，$i(n)$的计算如下：

$n = 16$，　$0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 = 0$

$i(1) \oplus i(6) \oplus i(8) \oplus i(9) \oplus i(10) \oplus i(14) = 0, i(16) = 0$



$n = 17,\quad 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 = 1$

$i(2) \oplus i(7) \oplus i(9) \oplus i(10) \oplus i(11) \oplus i(15) = 1, i(17) = 1$

$n = 18,\quad 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 = 0$

$i(3) \oplus i(8) \oplus i(10) \oplus i(11) \oplus i(12) \oplus i(16) = 0, i(18) = 0$

. ■.

. ■.

. ■.

这样可得到 $i(n)$，如表 3.5 所示。

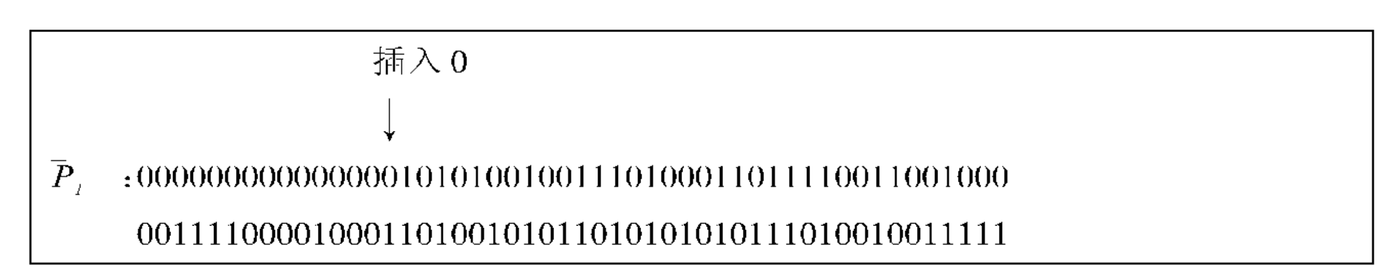
表 3.5　同相 PN 序列 $\bar{P}_I$ 的 $i(n)$ 生成

n	$\sum\limits_{k=15,10,8,7,6,2} i(n-k), 1 \leqslant n \leqslant 32\ 767$	$i(n)$
1		0
2		0
.		.
.		.
.		.
15	$i(n) = i(n-15) = 0, 1 \leqslant n \leqslant 14$	1
16		0
17		1
.		.
.		.
.		.
21	$0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1$	0
22	$0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0$	1
.		.
.		.
.		.
27	$0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1$	1
28	$0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1$	0
.		.
.		.
.		.
34	$1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0$	1
35	$0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1$	0
.		.
.		.
.		.
45	$0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1$	0
46	$0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0$	1
.		.
.		.
.		.

（续表）

n	$\sum_{k=15,10,8,7,6,2} i(n-k), 1 \leqslant n \leqslant 32\ 767$	$i(n)$
55	0⊕0⊕0⊕0⊕0⊕1	1
56	0⊕1⊕0⊕0⊕0⊕1	0
.		.
.		.
.		.
74	0⊕1⊕0⊕1⊕0⊕1	1
75	1⊕1⊕1⊕0⊕0⊕0	1
.		.
.		.
.		.
86	0⊕0⊕0⊕1⊕0⊕0	1
87	1⊕1⊕1⊕0⊕1⊕1	1
.		.
.		.
.		.
97	0⊕1⊕1⊕0⊕0⊕1	1
98	1⊕0⊕0⊕0⊕1⊕1	1
99	0⊕1⊕0⊕1⊕0⊕1	1
.		.
.		.
.		.

最后，在 $i(n)$连续输出 14 个 0 之后再插入 1 个 0（↓），得到如下的同相导频 PN 序列：


插入 0

↓

$\bar{P}_I$ ：00000000000000010101001001110100011011110011001000

00111100001000110100101011010101010111010010011111

类似地，可以得到正交相 PN 序列，如表 3.6 所示。

表 3.6　正交相 PN 序列 P_Q 的 $q(n)$生成

n	$\sum_{k=15,12,11,10,9,5,4,3} q(n-k), 1 \leqslant n \leqslant 32\ 767$	$q(n)$
1		0
2		0
.	$q(n) = q(n-15) = 0, 1 \leqslant n \leqslant 14$	.
.		.
.		.

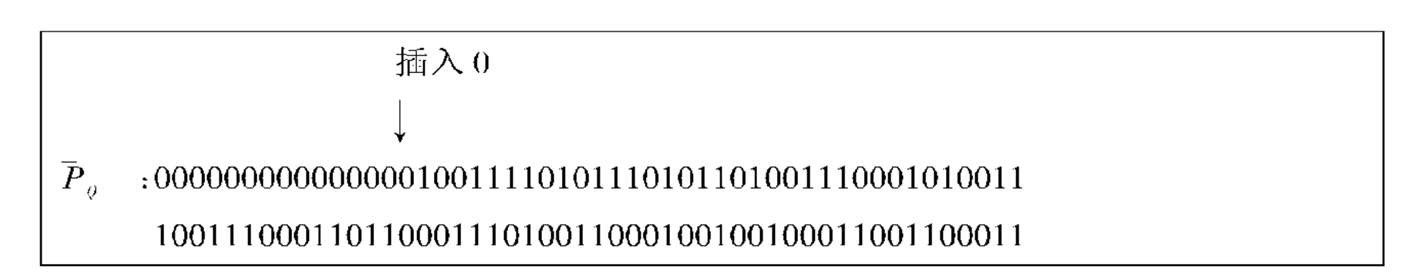
（续表）

n	$\sum_{k=15,12,11,10,9,5,4,3} q(n-k), 1 \leqslant n \leqslant 32\ 767$	$q(n)$
14		0
15		1
16	0⊕0⊕0⊕0⊕0⊕0⊕0⊕0	0
17	0⊕0⊕0⊕0⊕0⊕0⊕0⊕0	0
18	0⊕0⊕0⊕0⊕0⊕0⊕0⊕1	1
.		.
.		.
.		.
25	0⊕0⊕0⊕1⊕0⊕1⊕1⊕0	1
26	0⊕0⊕1⊕0⊕0⊕1⊕0⊕1	1
27	0⊕1⊕0⊕0⊕1⊕0⊕1⊕0	1
.		.
.		.
.		.
35	1⊕1⊕0⊕1⊕1⊕0⊕1⊕1	0
36	1⊕0⊕1⊕1⊕1⊕1⊕1⊕0	0
37	0⊕1⊕1⊕1⊕0⊕1⊕0⊕1	1
.		.
.		.
.		.
46	1⊕1⊕0⊕0⊕1⊕0⊕0⊕1	0
47	1⊕0⊕0⊕1⊕1⊕0⊕1⊕0	0
.		.
.		.
.		.
55	0⊕1⊕0⊕1⊕0⊕1⊕0⊕0	1
56	0⊕0⊕1⊕0⊕0⊕0⊕0⊕1	0
.		.
.		.
.		.
71	0⊕1⊕1⊕0⊕1⊕0⊕1⊕1	1
72	0⊕1⊕0⊕1⊕1⊕1⊕1⊕1	0
.		.
.		.
.		.
80	0⊕1⊕1⊕0⊕1⊕1⊕0⊕0	0
81	0⊕1⊕0⊕1⊕0⊕0⊕0⊕0	0

（续表）

n	$\sum_{k=15,12,11,10,9,5,4,3} q(n-k), 1 \leqslant n \leqslant 32\,767$	$q(n)$
.		.
.		.
.		.
88	$0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1$	0
89	$1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0$	1
.		.
.		.
.		.
97	$1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1$	0
98	$0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0$	1
99	$0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0$	1
.		.
.		.
.		.

利用表 3.6，在 $q(n)$ 连续输出 14 个 0 之后而插入 1 个 0（参见向下箭头），可以得到正交相位导频 PN 序列如下：


插入 0

↓

$\bar{P}_Q$ ：0000000000000001001111010111010110100111000101 0011

1001110001101100011101001100010010010001100110 0011

正如例 3.6 结尾所述，同相序列 I 和正交相位序列 Q 分别使用下面的公式，可以很容易计算 $I = \text{DSS} \oplus \bar{P}_I$ 和 $Q = \text{DSS} \oplus \bar{P}_Q$。

I 和 Q 数据都输入 I 和 Q 基带滤波器。但是数据脉冲 Q 应当延时半个 PN 码片时间（406.9 ns），以便运用偏移 QPSK。

3.1.8　基带滤波

正交相位扩展之后，即 $I = \text{DSS} \oplus \bar{P}_I$ 和 $Q = \text{DSS} \oplus \bar{P}_Q$，I 和 Q 路数据输入 I 和 Q 基带滤波器。令 $S(f)$ 是基带滤波器频谱。滤波器归一化频率响应在带通 $0 \leqslant f \leqslant f_p$ 内限制在 $\pm\delta_1$。归一化响应在 $f \geqslant f_s$ 时截止频率应该小于或等于 $-\delta_2$，如图 3.15 所示。特别地，限制参数的具体数值为：$\delta_1 = 1.5$ dB，$\delta_2 = 40$ dB，$f_p = 590$ kHz，$f_s = 740$ kHz。

令 $S(t)$ 为基带滤波器的冲击响应，则 $S(t)$ 满足下列方程：

$$\text{均方差} = \sum_{k=0}^{47} [\alpha S(kT_s - \tau) - h(k)]^2 \leqslant 0.03 \tag{3.9}$$

其中，α 和 τ 是最小化均方误差的常数。常数 T_s 等于 $1/4(1.2288 \times 10^6) = 203.451$ ns，刚好是一个 PN 码片时间的四分之一。

$k \geqslant 48$ 时，系数 $h(k) = h(47-k) = 0$，表 3.7 列出了 $k < 48$ 时的 $h(k)$ 的值。

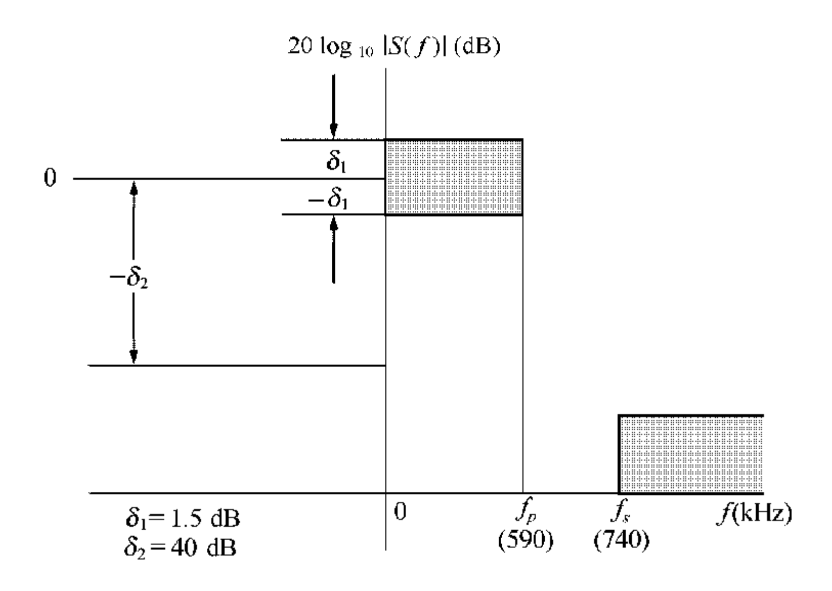


图 3.15 基带滤波器的频率响应限制(IS-95 之后)

表 3.7 系数 ***h*(*k*)**

k	***h*(*k*)**
0,47	−0.025■288■315
1,46	−0.034■167■931
2,45	−0.035■752■323
3,44	−0.016■733■702
4,43	0.021■602■514
5,42	0.064■938■487
6,41	0.091■002■137
7,40	0.081■894■974
8,39	0.037■071■157
9,38	−0.021■998■074
10,37	−0.060■716■277
11,36	−0.051■178■658
12,35	0.007■874■526
13,34	0.084■368■728
14,33	0.126■869■306
15,32	0.094■528■345
16,31	−0.012■839■661
17,30	−0.143■477■028
18,29	−0.211■829■088
19,28	−0.140■513■128
20,27	0.094■601■918
21,26	0.441■387■140
22,25	0.785■875■640
23,24	■ 1.0

(TIA/EIA/IS-95 之后)

3.1.9　偏移正交相移键控（OQPSK）

导频 PN 序列每 $2^{15}/1.2288\times10^{6}=26.666$ ms 重复一次。因此，每 2 秒共有 75 次重复（即 $2/26.666\times10^{-3}=75$）。由于 Q 导频 PN 序列扩展的数据延时了半个码片时间，所以使用偏移正交相移键控扩展调制，这在以正交相位于 2 个载波上同时传输时很有优势。

OQPSK 和标准的非偏移 QPSK 在两个基带波形对齐方面不同。令 $I(t)$和 $Q(t)$为基带滤波器输出流。由于 Q 延时半个码片时间，脉冲流 $I(t)$和 $Q(t)$定时偏移 $T_b/2$ 秒。在标准非偏移 QPSK 下，两种脉冲流时间一致，载波相位每 T_b 秒变化一次。

将每个 $I(t)$和 $Q(t)$幅度调制到载波的余弦和正弦函数上，可以得到正交 QPSK 波形$s(t)$，如图 3.16 所示。

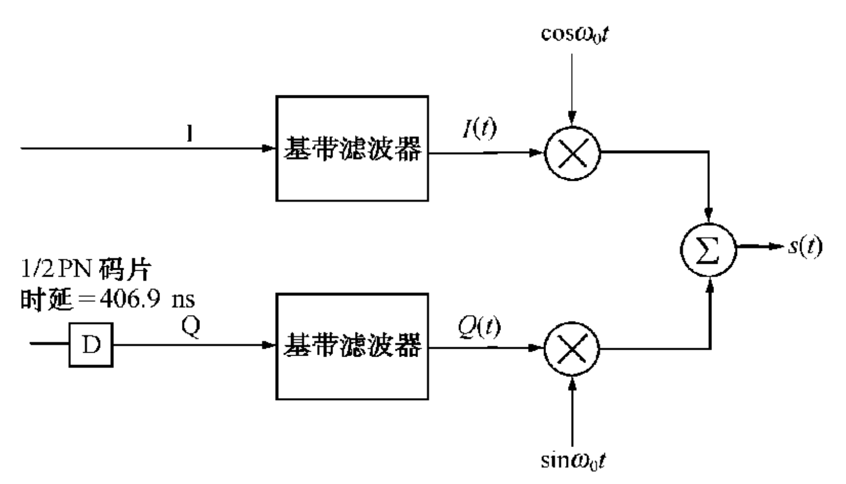


图 3.16　幅度调制 OQPSK 波形的产生

同相流 $I(t)$以 +1 或 −1 对余弦函数进行幅度调制，产生一个 BPSK 波形。同样，正交相位流 $Q(t)$对正弦函数进行调制，产生另一个与前面余弦函数产生波形正交的 BPSK 波形。这样，这两个正交 BPSK 波形的和就产生 QPSK 波形。

令 $s(t)$为 QPSK 波形，表达式为：

$$s(t)=I(t)\cos\omega_0 t+Q(t)\sin\omega_0 t \tag{3.10}$$

将 $I(t)=\sqrt{2}\cos\theta(t)$和 $Q(t)=\sqrt{2}\sin\theta(t)$代入后，$s(t)$变为：

$$s(t)=\sqrt{2}\cos(\omega_0 t-\theta(t)) \tag{3.11}$$

其中 $\theta(t)=\arctan\dfrac{Q(t)}{I(t)}$

$\theta(t)$的值对应于 $I(t)$和 $Q(t)$可能的四种组合之一。对应 $I(t)$和 $Q(t)$特定值的 QPSK 输出 $s(t)$，可根据 $\theta(t)$的值确定，如下所示：

1. $\theta(t)=\dfrac{\pi}{4}$时，$I(t)$和 $Q(t)$变为 $I(t)=\sqrt{2}\cos\left(\dfrac{\pi}{4}\right)=1$，

   $Q(t)=\sqrt{2}\sin\left(\dfrac{\pi}{4}\right)=1$，$s(t)=\sqrt{2}\cos\left(\omega_0 t-\dfrac{\pi}{4}\right)$

2. $\theta(t)=\dfrac{3\pi}{4}$时，$I(t)=\sqrt{2}\cos\left(\dfrac{3\pi}{4}\right)=-1$，$Q(t)=\sqrt{2}\sin\left(\dfrac{3\pi}{4}\right)=1$，

   $s(t)=\sqrt{2}\cos\left(\omega_0 t-\dfrac{3\pi}{4}\right)$

3. $\theta(t)=-\frac{3\pi}{4}$时，$I(t)=\sqrt{2}\cos\left(-\frac{3\pi}{4}\right)=-1$，

$Q(t)=\sqrt{2}\sin\left(-\frac{3\pi}{4}\right)=-1$，$s(t)=\sqrt{2}\cos\left(\omega_0 t+\frac{3\pi}{4}\right)$

4. $\theta(t)=-\frac{\pi}{4}$时，$I(t)=\sqrt{2}\cos\left(-\frac{\pi}{4}\right)=1$，

$Q(t)=\sqrt{2}\sin\left(-\frac{\pi}{4}\right)=-1$，$s(t)=\sqrt{2}\cos\left(\omega_0 t+\frac{\pi}{4}\right)$

基于以上分析，反向 CDMA 信道（接入信道或反向业务信道）的 $I(t)$ 和 $Q(t)$ 映射如表 3.8 所示。

表 3.8 接入信道（反向业务信道）的 $I(t)$ 和 $Q(t)$ 映射

$\theta(t)$	$I(t)$	$Q(t)$		$I(t)$	$Q(t)$
$\pi/4$	1	1		0	0
$3\pi/4$	–1	1		1	0
$-3\pi/4$	–1	–1	⇨	1	1
$-\pi/4$	1	–1		0	1
	(不归 0 值)			(二进制值)	

利用表 3.8，可以得到图 3.17 所示的信号星座映射和相位转移。

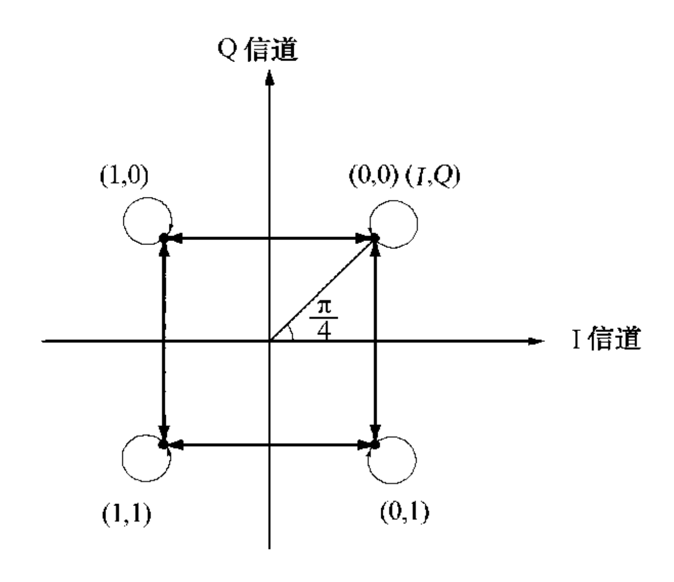


图17 接入信道或反向业务信道的信号星座和相位转移

利用图 3.9 所示的 $(I(t), Q(t), \theta(t))$ 映射可以画出 QPSK 脉冲流 $s(t)=\sqrt{2}\cos(\omega_0 t-\theta(t))$。将表 3.9 作为工具，可以得到图 3.18 所示的 QPSK 脉冲，它是描绘 $s(t)$ 的重要元素。

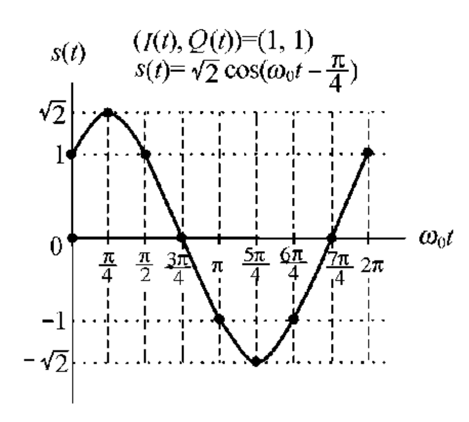
表 3.9 QPSK 波形图数据表

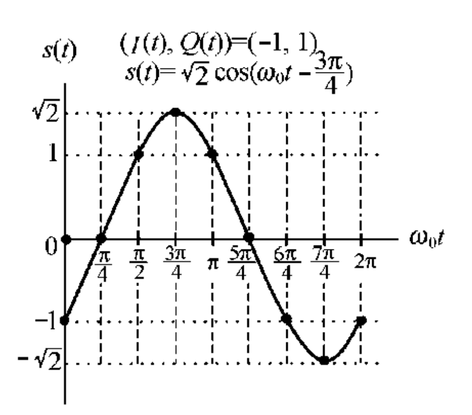
$\omega_0 t$	$s(t)=\sqrt{2}\cos(\omega_0 t-\theta(t))$			
	$(1,1), \theta(t)=\pi/4$	$(-1,1), \theta(t)=3\pi/4$	$(-1,-1), \theta(t)=-3\pi/4$	$(1,-1), \theta(t)=-\pi/4$
0	1	–1	–1	1

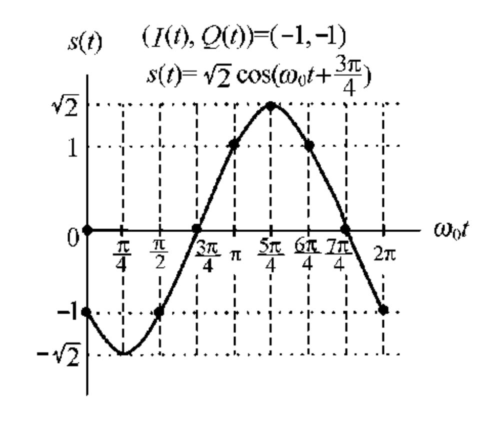
$\pi/4$	$\sqrt{2}$	0	$-\sqrt{2}$	0

（续表）

$\omega_0 t$	$s(t)=\sqrt{2}\cos(\omega_0 t-\theta(t))$			
	$(1,1),\theta(t)=\pi/4$	$(-1,1),\theta(t)=3\pi/4$	$(-1,-1),\theta(t)=-3\pi/4$	$(1,-1),\theta(t)=-\pi/4$
$\pi/2$	1	1	−1	−1
$3\pi/4$	0	$\sqrt{2}$	0	$-\sqrt{2}$
π	−1	1	1	−1
$-3\pi/4$	$-\sqrt{2}$	0	$\sqrt{2}$	0
$-\pi/2$	−1	−1	1	1
$-\pi/4$	0	$-\sqrt{2}$	0	$\sqrt{2}$
2π	1	−1	1	1







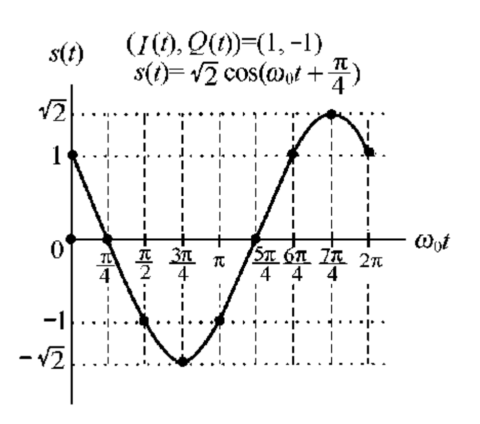


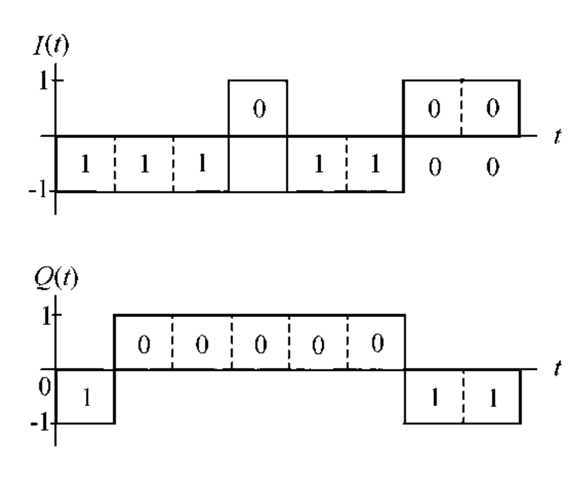
图 3.18 根据（$I(t)$，$Q(t)$，$\theta(t)$）在 $T_b/2$ 上映射的 QPSK 脉冲图解

例 3.10 正如前面计算过的，同相序列 I 和正交相序列 Q 分别为：

I：	10010011	00101101	11101100	11100000	11100111	…
Q：	10010011	00101101	10000011	11100001	00101111	…

出于演示的目的，只考虑部分序列 $I(t)$ =（11101100）和 $Q(t)$ =（10000011）。图 3.19 分别显示了 $I(t)$ 和 $Q(t)$ 以及 QPSK 波形。

类似地，图 3.20 显示了 Q 延时半个码片 $T_b/2$ 的偏移 QPSK 脉冲流。



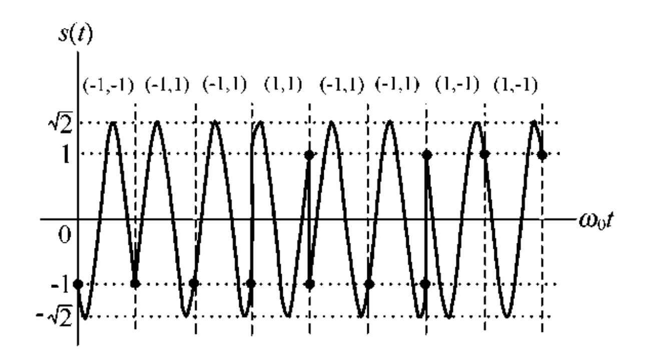
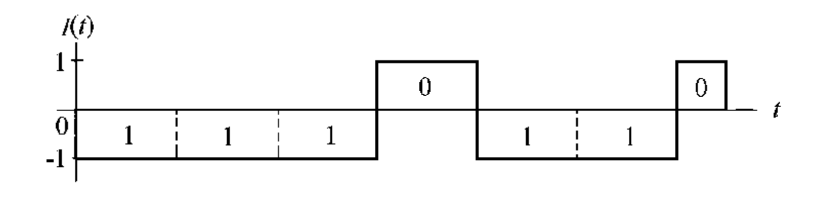
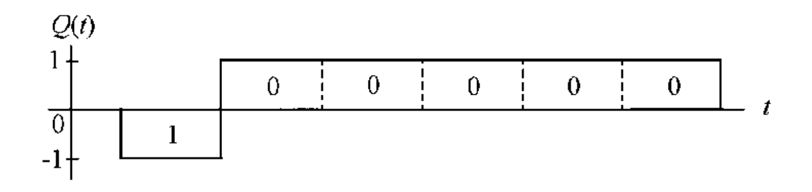


图 3.19　对应于 $I(t)$ = (11101100)和 $Q(t)$ = (10000011)的 QPSK 波形





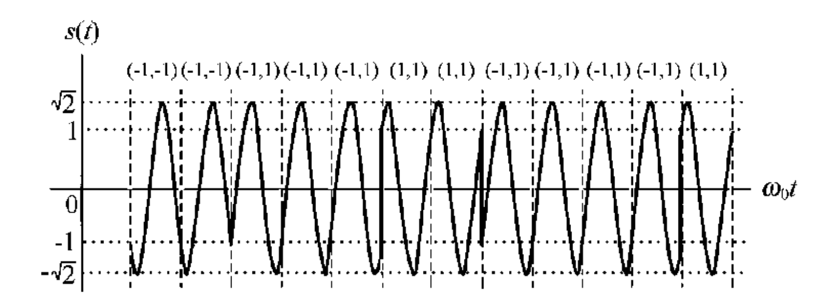


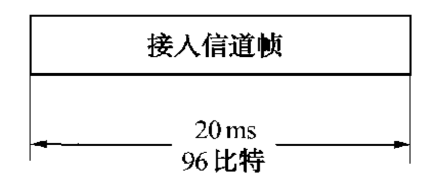
图 3.20　对应于 $I(t)$ =(11101100)和 $Q(t)$ =(10000011)的 OQPSK 波形 $s(t)$

3.1.10　接入信道报头和消息封装

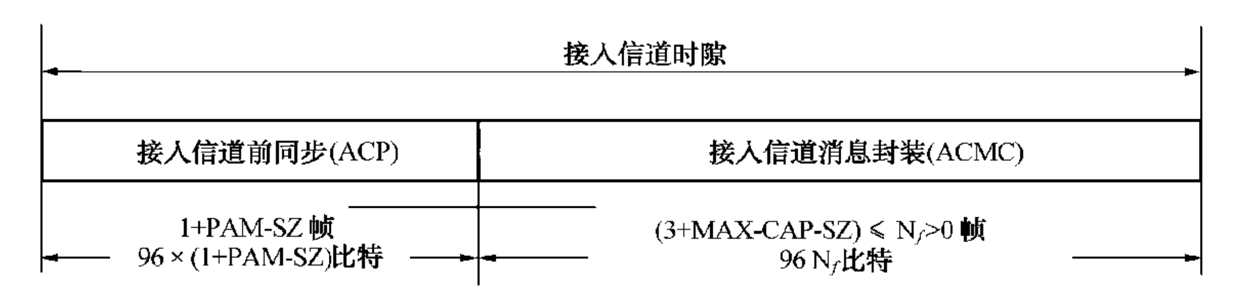
反向 CDMA 信道支持的每个寻呼信道可以含有从编号 0 到编号 31 的 32 路接入信道。每路接

入信道和单一寻呼信道相关联。移动台使用接入信道发起和基站的通信并相应寻呼信道消息。固定数据率为 4.8 kb/s 的接入信道帧在 20 ms 持续时间内包含 96 比特，帧开始于系统时间为 20 ms 的整数倍的时刻。

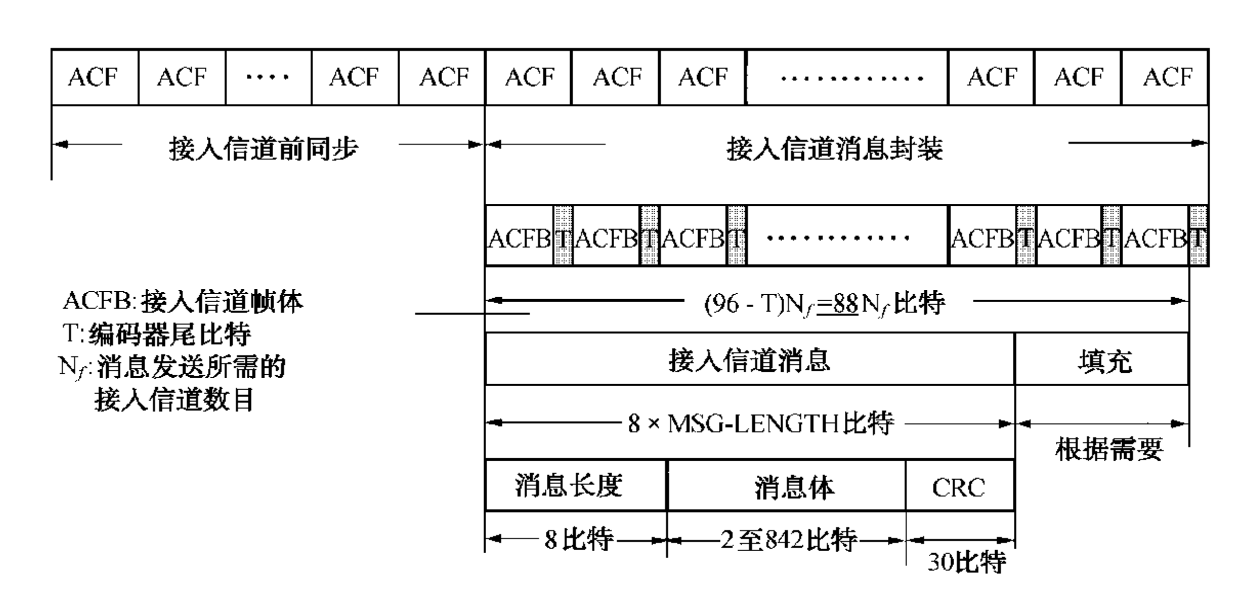
接入信道发送包括接入信道报头和接入信道消息封装。发送接入信道报头的作用是帮助基站捕获接入信道传输。接入信道消息封装包括接入信道消息和填充，如图 3.21(c)所示。接入信道消息包括消息长度字段(8 比特)、消息体(2 到 842 比特)和以图中顺序出现的 CRC(30 比特)，但不包括报头或填充。



(a)接入信道帧



(b)接入信道时隙



(c)接入信道消息封装＝接入信道消息＋填充

图 3.21 接入信道结构

接入信道消息封装的长度为接入信道帧的整数个数，由下列公式决定：

$$\text{封装大小} = \left\lceil \frac{\text{消息长度} + \text{消息体长度} + \text{CRC}}{\text{无编码器尾比特的消息帧}} \right\rceil$$

$$=\left\lceil \frac{8+(2\text{ 至 }842)+30}{88} \right\rceil \qquad (3.12)$$

消息体大小的选取满足封装大小不超过3 + MAX-CAP-SZ(一个接入信道消息封装内最大的接入信道帧数)。

移动台发送的填充包含0或多个0比特,它紧跟接入信道消息。填充的长度满足式(3.13)。

$$8+\text{消息体长度}+30+\text{填充长度} = 88\times\text{封装大小} \qquad (3.13)$$

例3.11　确定消息体长度为460比特时,封装大小和填充长度为:

$$\text{封装大小}=\left\lceil \frac{8+460+30}{88} \right\rceil=\left\lceil \frac{498}{88} \right\rceil=6$$

填充长度 = (88 × 6) − 498 = 528 − 498 = 30比特(全0)。

由于封装大小不能超过$3+N_{mf}$,且因为$6\leqslant 3+N_{mf}$,所以接入信道帧N_{mf}的最大数至少是3。利用式(3.12)和式(3.13),可以确定对应于消息体长度的封装大小和接入信道帧数,如表3.10所示。

表3.10　封装大小和接入信道帧数目的确定

消息体长度	封装大小	接入信道帧数
227≤MBL≤314	4	1
315≤MBL≤402	5	2
403≤MBL≤490	6	3
491≤MBL≤578	7	4
579≤MBL≤666	8	5
667≤MBL≤754	9	6
755≤MBL≤842	10	7

利用表3.10,可以得到图3.22所示的封装大小相对消息体长度的曲线和接入信道帧数相对封装大小的曲线。

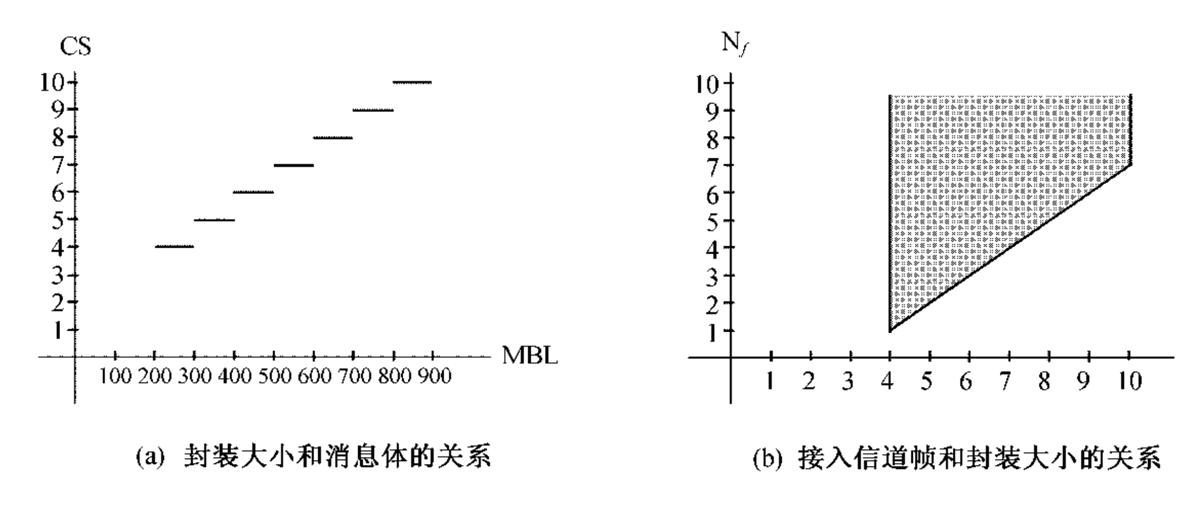


图3.22

3.1.11　接入信道CRC计算

每个接入信道信令消息要计算30位CRC。CRC序列由消息长度字段和消息体共同产生。

CRC 生成多项式如下：

$$g(x) = 1 + x + x^2 + x^6 + x^7 + x^8 + x^{11} + x^{12} + x^{13} + x^{15} + x^{20} + x^{21} + x^{29} + x^{30} \quad (3.14)$$

逻辑示意图如图 3.23 所示。

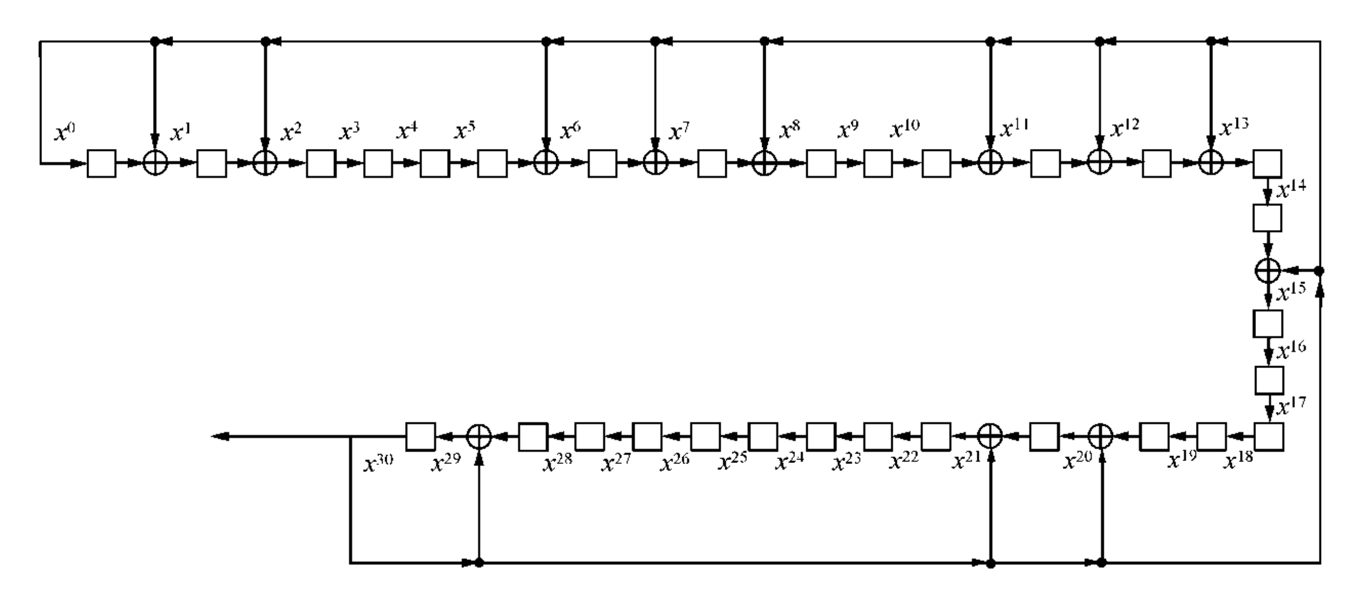


图 3.23 方程 3.14 中 $g(x)$ 表示的 LFSR 电路

CRC 字段可以按照下列步骤计算，相应的 CRC 编码器如图 3.24 所示。

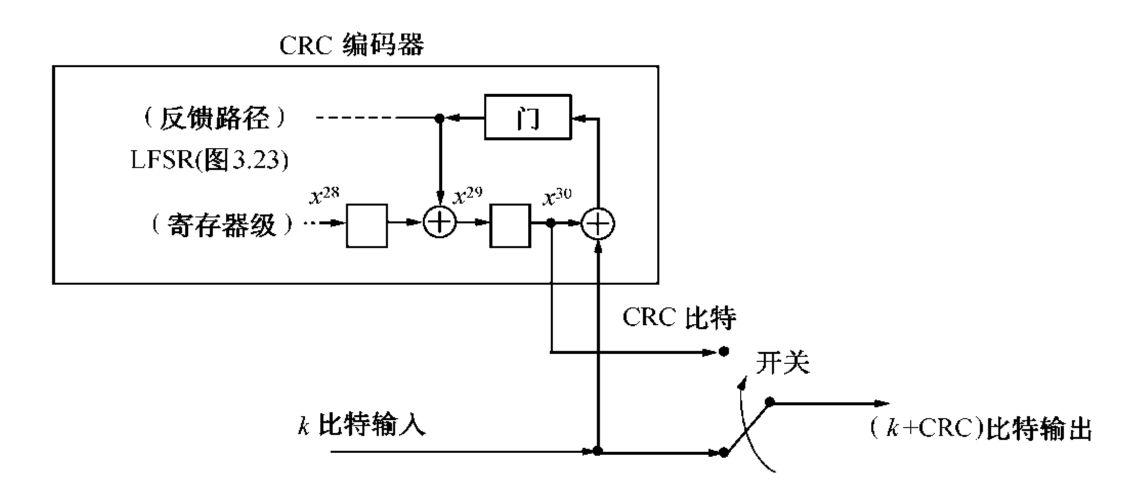


图 3.24 CRC 字段产生的编码器框图

1. 移位寄存器初始内容设置为全 1，以便即使数据为全 0 时 CRC 字段为非 0。
2. 将开关置于向下位置，关闭 CRC 编码器反馈路径中的门电路。
3. k 输入比特开始输入编码器输出与移位寄存器反馈路径。k 比特输入表示 8 + 消息体长度的比特。
4. 寄存器每输入 k 比特就锁定 k 次。
5. 将开关置于向上位置，打开反馈路径中的门电路。
6. 寄存器额外计时 30 次，将产生 CRC 字段。
7. 这样，数据比特将在 CRC 编码器输出以消息长度字段、消息体长度和 CRC 字段的顺序输

出。

移动台将接受那些CRC校验有效的消息,那些无效的消息将被忽略。

3.1.12　接入信道调制参数

到目前为止,我们已经涉及到接入信道的详细内容。总结一下,所有在接入信道发送的数据采取下列步骤处理:

1. 比特后添加8位编码尾比特以封装成20 ms帧。
2. 数据进行卷积编码。
3. 卷积编码器输出的编码符号在进行交织之前重复。
4. 使用块交织的目的是防止调制之前发送数据时发生突发错误。
5. 在64阶正交调制时,调制符号是由Walsh函数产生的64个互相正交的波形之一。
6. 发送之前运用长码直接序列扩展。扩展操作包括64阶正交输出和长码的模2加。
7. 直接序列扩展之后,接入信道正交相位扩展。正交相位数据序列 I 和 Q 是由0偏移I和Q导频PN序列和长码扩展直接序列的输入数据进行"异或"操作得到的序列。
8. Q导频PN序列扩展的正交相位序列数据 Q 延时半个码片时间(406.9 ns)。
   这些 I 和 Q 脉冲输入I和Q基带滤波器。基带滤波之后,I 和 Q 数据脉冲映射到相位转移。
9. 将每个 $I(t)$ 和 $Q(t)$ 幅度调制到载波的余弦和正弦函数上去可以得到正交QPSK波形 $s(t)$。由于Q导频PN序列延时半个码片时间,所以使用偏移正交相移键控(OQPSK)进行扩展调制。

接入信道调制参数如表3.11所示。

表3.11　接入信道调制参数

参　数	数　据　率	单　　位
PN码片速率	1.228 8	Mc/s
码率	1/3	比特/码符号
码符号重复	2	符号/码符号
发送负载周期	100.0	%
码符号速率	28 800	s/s
调制	6	码符号/调制符号
调制符号速率	4 800	s/s
Walsh码片速率	307.20	kc/s
调制符号持续时间	208.33	μs
PN码片/码符号	42.67	PN码片/码符号
PN码片/调制符号	256	PN码片/调制符号
PN码片/Walsh码片	4	PN码片/Walsh码片

3.2　反向业务信道

反向业务信道(RTC)用于呼叫过程中向基站传输用户数据和信令信息。反向信道可使用9.6 kb/s,4.8 kb/s,2.4 kb/s或1.2 kb/s四种不同速率。

RTC 数据率基于帧来选择。帧表示 CDMA 系统中的基本定时间隔。对反向业务信道而言，一帧长 20 ms。

反向业务信道中发送占空比随着传输的数据率变化。特别地，传输速率为 9.6 kb/s，4.8 kb/s，2.4 kb/s 和 1.2 kb/s 的帧占空比分别为 100%，50%，25% 和 12.5%。

每路反向业务信道由不同的用户长码序列标识。长码序列在反向业务信道和前向业务信道惟一标识移动台。

反向业务信道具有图 3.25 所示的完整结构。反向业务信道发送的所有数据经过卷积编码、块交织、64 阶正交调制和直接序列扩展再发送。

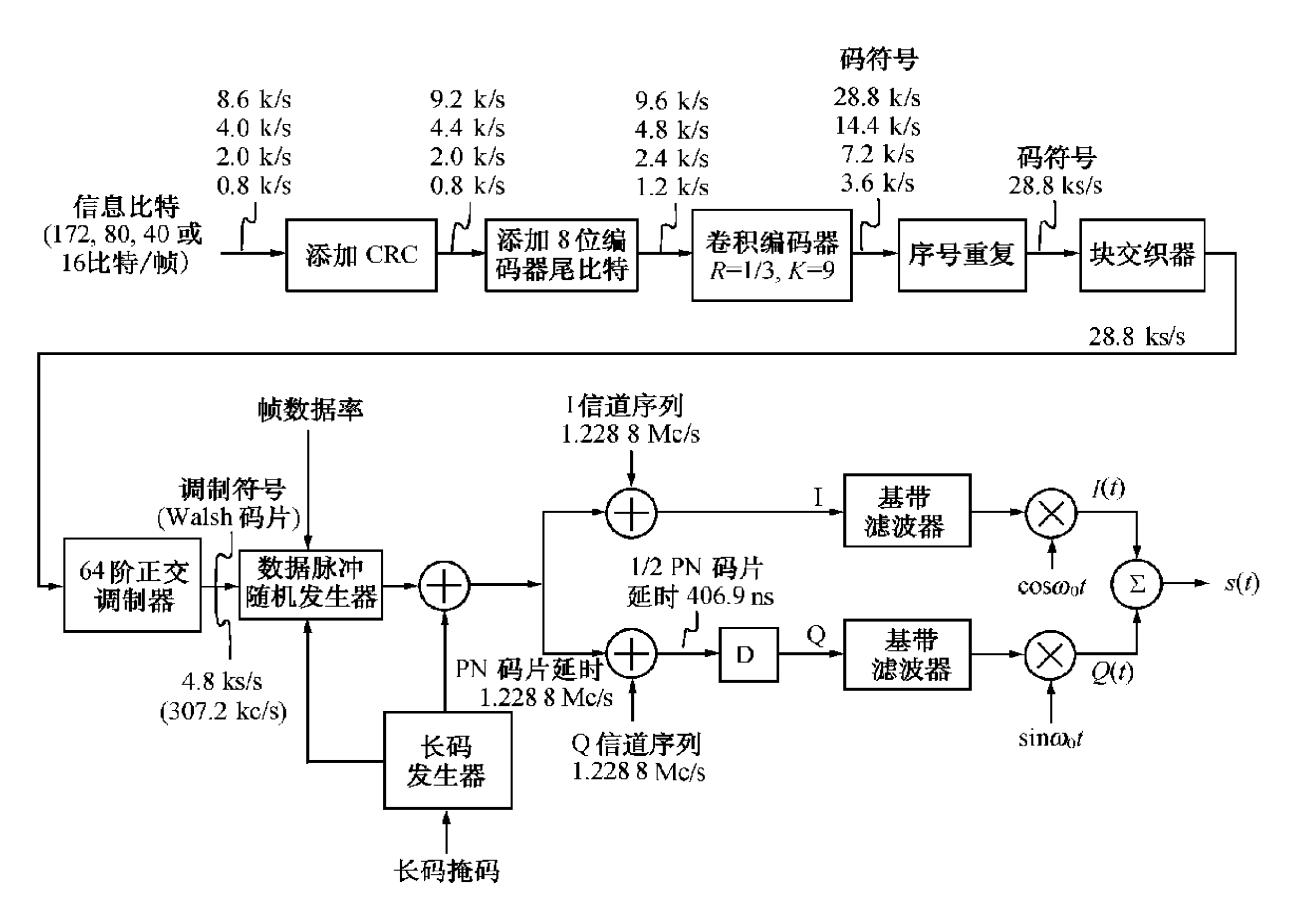


图 3.25 反向业务信道结构

参见图 3.25，9.6 kb/s 和 4.8 kb/s 发送数据率的计算按照下列两步进行：

1. 9.2 kb/s 和 4.4 kb/s 的数据率是在 8.6 kb/s 和 4.0 kb/s 信息速率加上帧质量指示（即 9.2 kb/s 速率加上 12 位 CRC，4.4 kb/s 速率加上 8 位 CRC）计算得到的。
2. 加上 8 位编码尾比特，将由 9.2 kb/s 和 4.4 kb/s 的数据率计算得到 9.6 kb/s 和 4.8 kb/s 发送数据率。参见表 3.12，速率为 9.6 kb/s 时帧长 192 比特，发送速率由 8.6 kb/s 到 9.6 kb/s 的计算方式为：192 比特/20 ms = 9.6 kb/s。类似地，速率为 4.8 kb/s 时帧长 96 比特，发送速率由 4.0 kb/s 到 4.8 kb/s 的计算方式为：96 比特/20 ms = 4.8 kb/s。

接着，信息速率 2.0 kb/s，0.8 kb/s 分别添加 8 位编码尾比特可以计算得到 2.4 kb/s 和 1.2 kb/s 的发送速率。因此，2.0 kb/s 信息速率到 2.4 kb/s 由 48 比特/20 ms = 2.4 kb/s 计算得到；类似地，0.8 kb/s 信息速率到 1.2 kb/s 由 24 比特/20 ms = 1.2 kb/s 计算得到。

3.2.1 RTC 帧结构

每个 RTC 数据帧持续 20 ms，根据发送速率不同可能含有信息、CRC、尾比特或信息、尾比特，具体如下：

1. 以 9.6 kb/s 和 4.8 kb/s 速率发送的 RTC 帧分别包含 192 比特和 96 比特。

   对于 9 600 b/s

   帧比特 = 172 信息比特 +12 CRC 比特 +8 编码尾比特 = 192 比特

   对于 4 800 b/s

   帧比特 = 80 信息比特 +8 CRC 比特 +8 编码尾比特 = 96 比特

2. 以 2.4 kb/s 和 1.2 kb/s 速率发送的 RTC 帧分别包含 48 比特和 24 比特。

   对于 2 400 b/s

   帧比特 = 40 信息比特 +8 编码尾比特 = 48 比特

   对于 1 200 b/s

   帧比特 = 16 信息比特 +8 编码尾比特 = 24 比特

表 3.12 **RTC** 帧结构

数据率(b/s)	信息比特	CRC 比特	编码尾比特	帧长(比特)
9 600	172	12	8	192
4 800	80	8	8	96
2 400	40	–	8	48
1 200	16	–	8	24

3.2.2 调制参数和特性

反向业务信道可能使用四种数据率之一来发送数据，但是发送占空比随着发送数据率变化，如表 3.13 所示。特别地，传输速率为 9.6 kb/s，4.8 kb/s，2.4 kb/s 和 1.2 kb/s 的帧占空比分别为 100%，50%，25% 和 12.5%。然而，实际编码符号速率固定在 28.8 ks/s。传输速率为 9.6 kb/s，4.8 kb/s，2.4 kb/s 或 1.2 kb/s 时，反向业务信道调制是 64 阶正交调制。每 6 个编码符号对应发送 64 种可能的调制符号之一。该调制符号是 Walsh 函数产生的 64 个相互正交波形之一。由于块交织器输出的码符号速率固定在 28.8 ks/s，固定 Walsh 码片速率变成 28.8/6 = 4.8 ks/s 或 4.8 × 64 = 307.2 kc/s。每个 Walsh 码片由 4 个 PN 码片扩展，因为 PN 码片和 Walsh 码片速率比是 1228.8/307.2 = 4。扩展 PN 码片序列速率固定为 1.228 8 Mc/s。调制符号持续时间为 1/4800 = 208.33 μs。这样，表 3.13 定义了反向业务信道不同传输速率信号速率和它们之间的关系。

表 3.13 反向业务信道调制参数

参数	数据率				单位
	9 600	**4 800**	**2 400**	**1 200**	
PN 码片速率	1.228 8	1.228 8	1.228 8	1.228 8	Mc/s
码率	1/3	1/3	1/3	1/3	比特/码符号
发送负载周期	100	50	25	12.5	%
码符号率	28 800	28 800	28 800	28 800	s/s

（续表）

参数	数据率				单位
	9 600	4 800	2 400	1 200	
调制	6	6	6	6	码符号/调制符号
调制符号速率	4 800	4 800	4 800	4 800	s/s
Walsh 码片率	307.20	307.20	307.20	307.20	kc/s
调制符号持续时间	208.33	208.33	208.33	208.33	μs
PN 码片/码符号	42.67	42.67	42.67	42.67	PN 码片/码符号
PN 码片/调制符号	256	256	256	256	PN 码片/调制符号
PN 码片/Walsh 码片	4	4	4	4	

反向业务信道完整结构如图 3.25 所示。

3.2.3 RTC 帧质量指示器

在四种不同的帧传输速率中，每个 9 600 b/s 和 4 800 b/s 帧都包含一个帧质量指示器（CRC）。传输速率为 2 400 b/s 和 1 200 b/s 时，不使用帧质量指示器。

对于 9 600 b/s 和 4 800 b/s 速率，CRC 的计算是基于帧内除了 CRC 比特和尾比特之外的信息比特进行的。计算 9 600 b/s CRC 的生成多项式为：

$$g(x) = 1 + x + x^4 + x^8 + x^9 + x^{10} + x^{11} + x^{12} \tag{3.15}$$

其逻辑图如图 3.26 所示。

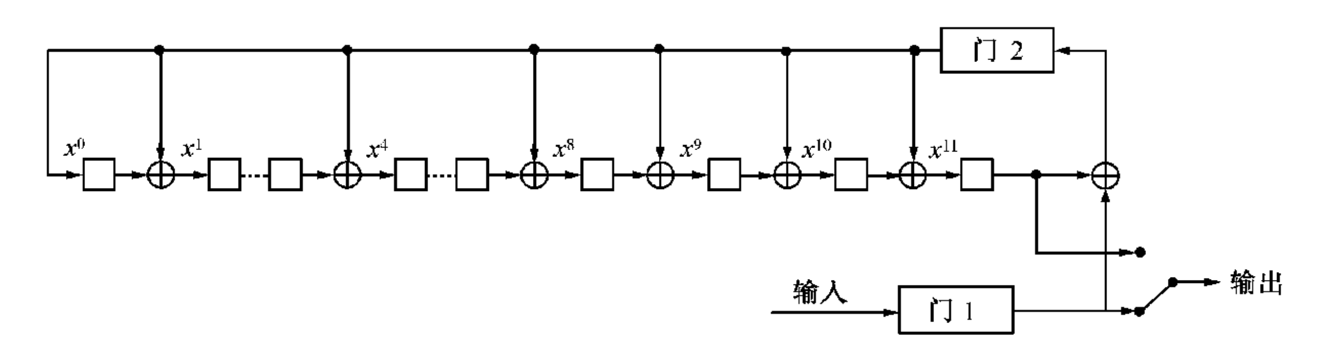


图 3.26 9 600 b/s 数据率的 CRC 计算电路

计算 4 800 b/s CRC 的生成多项式为：

$$g(x) = 1 + x + x^3 + x^4 + x^7 + x^8 \tag{3.16}$$

其 LFSR 电路如图 3.27 所示。

利用图 3.26 和 3.27，根据下列过程计算帧质量指示器：

1. 将所有移位寄存器状态设置为逻辑 1，开关置于下面的位置，门 1 和门 2 处于选通状态。
2. 对于 192 比特帧（20 ms），寄存器锁定 172 次，96 比特帧锁定 80 次。
3. 读入全部信息比特，关闭门 1。
4. 开关置于上面位置之后，对于 192 比特帧，寄存器再额外锁定 12 次，96 比特帧再锁定 8 次。这 12 位或 8 位额外输出比特就是 CRC 比特。
5. 这些 CRC 比特将以计算顺序发送。

每个 RTC 帧的最后 8 比特，称为编码尾比特，将被设置为逻辑 0。

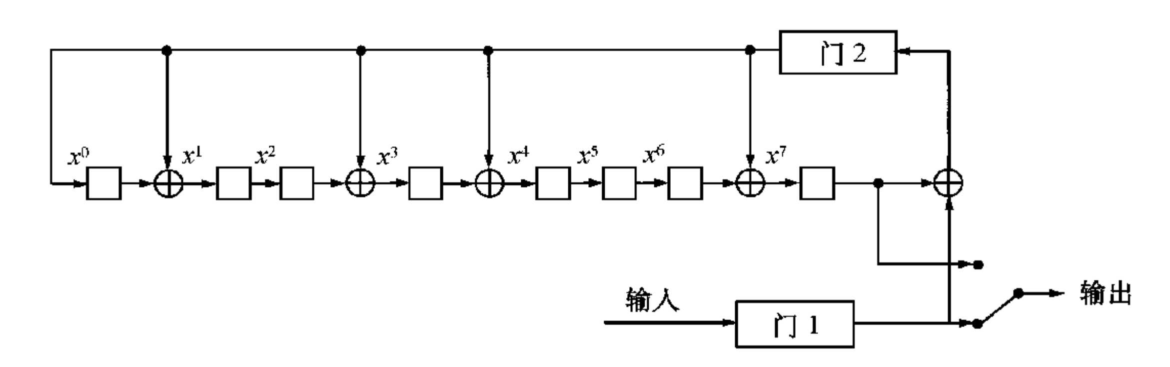


图 3.27　4 800 b/s 数据率的 CRC 计算电路

例 3.12　考虑 4 800 b/s 帧(20 ms)的 80 位信息数据序列如下:

1010110011 1001101111 0010010100 0100110011 0011001000 1110010110 1001100110 0110011110

寄存器对 96 比特帧锁定 80 次。利用图 3.27,计算可得表 3.14 所示的 CRC 比特。

表 3.14　寄存器初始内容为(11111111)的 4 800 b/s 帧的 CRC 计算

移位编号	信息比特	反馈比特	寄存器内容
1	1	1 ⊕ 1 = 0	01111111
2	0	0 ⊕ 1 = 1	11100110
3	1	1 ⊕ 0 = 1	10101010
4	0	0 ⊕ 0 = 0	01010101
5	1	1 ⊕ 1 = 0	00101010
6	1	1 ⊕ 0 = 1	11001100
.	.	.	.
21	0	0 ⊕ 1 = 1	10000011
22	0	0 ⊕ 1 = 1	10011000
23	1	1 ⊕ 0 = 1	10010101
.	.	.	.
35	1	1 ⊕ 1 = 0	00000000
36	1	1 ⊕ 0 = 1	11011001
37	0	0 ⊕ 1 = 1	10110101
.	.	.	.
50	0	0 ⊕ 0 = 0	00010111
51	1	1 ⊕ 1 = 0	00001011
52	1	1 ⊕ 1 = 0	00000101
.	.	.	.
78	1	1 ⊕ 1 = 0	00110000
79	1	1 ⊕ 0 = 1	11000001
80	0	0 ⊕ 1 = 1	10111001

参见表 3.14,$n = 80$ 时,寄存器内容为 10111001。当寄存器额外锁定了 8 次时,寄存器输出将成为 CRC 比特。进行 CRC 计算后的信息序列变成:

1010110011 1001101111 0010010100 0100110011 0011001000 1110010110 1001100110 0110011110 10011101

最后,仅仅加上编码尾比特(额外全 0 的 8 比特)就可以得到 4 800 b/s 速率的 RTC 帧序列,表

示如下。

1010110011 1001101111 0010010100 0100110011 0011001000 1110010110 1001100110 0110011110 1001110100 000000

（Ⅰ）

这个帧序列就是(3，1，8)卷积编码器的输入。

例 3.13　假设 9 600 b/s 帧(20 ms)的 172 位信息数据序列如下：

1100100011 0111001111 0011010110 0100110011 0011001111 0011010110 1110110011 0011001111 0011010110 1001100110 0110011001 1110011011 0001010010 0110011001 1011001010 1001100110 0110011110 01

利用图 3.27，移位寄存器对 192 比特帧锁定 172 次，计算可得表 3.15 所示的 CRC。

表 3.15　寄存器初始内容设置为(111111111111)的 9 600 b/s 帧的 CRC 计算

位移编号	信息比特	反馈比特	寄存器内容
1	1	$1 \oplus 1 = 0$	011111111111
2	1	$1 \oplus 1 = 0$	001111111111
3	0	$0 \oplus 1 = 1$	110101110000
.	.	.	.
80	1	$1 \oplus 0 = 1$	100000010110
81	0	$0 \oplus 0 = 0$	010000001011
82	0	$0 \oplus 1 = 1$	111010001010
.	.	.	.
125	0	$0 \oplus 1 = 1$	111101001100
126	1	$1 \oplus 0 = 1$	101100101001
127	0	$0 \oplus 1 = 1$	100100011011
.	.	.	.
170	0	$0 \oplus 0 = 0$	000011111100
171	0	$0 \oplus 0 = 0$	000001111110
172	1	$1 \oplus 0 = 1$	110010110000

参见表 3.15 中，$n = 172$ 时，寄存器内容将成为额外锁定 12 次时产生的 CRC 比特。这样，信息加上 CRC 就变成：

1100100011 0111001111 0011010110 0100110011 0011001111 0011010110 1110110011 0011001111 0011010110 1001100110 0110011001 1110011011 0001010010 0110011001 1011001010 1001100110 0110011110 0100001101 0011…

最后，仅仅加上编码尾比特就可以得到 9 600 b/s 速率的 RTC 帧序列，表示如下。

1100100011 0111001111 0011010110 0100110011 0011001111 0011010110 1110110011 0011001111 0011010110 1001100110 0110011001 1110011011 0001010010 0110011001 1011001010 1001100110 0110011110 0100001101 0011000000 000000

（Ⅱ）

这个帧序列就是(3，1，8)卷积编码器的输入。

3.2.4　$R = 1/3, K = 9$ 的卷积编码器

移动台对反向业务信道上发送的帧数据序列进行卷积编码再进行块交织。相同的(3,1,8)卷积编码器(图 3.14)同样适用于接入信道和反向业务信道。该编码器的生成序列为 $g_1^{(1)}$ = (101101111),$g_1^{(2)}$ = (110110011),$g_1^{(3)}$ = (111001001)。

卷积编码器的初始状态假设为全 0。编码器每输入 1 位编码输入序列,将产生 3 位编码符号(c_0, c_1, c_2)。

卷积编码包括连续时延数据序列的选定抽头的模 2 加法运算。数据序列时延长度等于 $m = K - 1 = 8$,其中 K 表示码的约束长度。图 3.4 显示了反向业务信道的卷积编码器。

例 3.14　传输速率为 1 200 b/s 的 RTC 帧包含 24 比特,这 24 比特由 16 位信息比特加上 8 为尾比特组成。编码尾比特设置成逻辑 0。假设 16 位信息比特为 1110001011010011,8 位编码尾比特为 00000000。

卷积编码器输入帧序列变为 111000101101001100000000。由于码率 $R = 1/3$,对于每位输入帧比特,编码器产生三位编码符号(c_0, c_1, c_2)。

利用图 3.4,可以计算得到表 3.16 中的三位编码符号。

表 3.16　利用图3.4产生的三位编码符号(c_0, c_1, c_2)

移位编号	输入比特	寄存器内容	输出码符号
1	1	00000000	111
2	1	10000000	100
3	1	11000000	001
4	0	11100000	000
5	0	01110000	001
6	0	00111000	001
7	1	00011100	100
8	0	10001110	100
9	1	01000111	111
10	1	10100011	011
11	0	11010001	011
12	1	01101000	001
13	0	10110100	011
14	0	01011010	100
15	1	00101101	111
16	1	10010110	100
17	0	11001011	010
18	0	01100101	000
19	0	00110010	010
20	0	00011001	000
21	0	00001100	001
22	0	00000110	010
23	0	00000011	001
24	0	00000001	111

这样,每个模 2 加法器产生的编码符号分别为:

$$c_0 = (110000111000011100000001)$$

$$c_1 = (100000001110101010100101)$$
$$c_2 = (101011001111101000001011)$$

最后，通过转换开关并按列位移 c_0, c_1, c_2，可以产生编码符号输出序列。

1111000010 0000100110 0100111011 0110010111 0011110001 0000010000 0010100011 11	(Ⅲ)

例 3.15 重新考虑传输速率为 1 200 b/s RTC 的(3，1，8)卷积编码器输出序列的计算。下面将看到编码器输出的另几种方案。

令 d 为带有尾比特的信息序列：

$$d = (1110\ 0010\ 1101\ 0011\ 0000\ 0000)$$

生成序列：

$$g_1^{(1)} = (557) = (101\ 101\ 111) = D^8 + D^6 + D^5 + D^3 + D^2 + D + 1$$
$$g_1^{(2)} = (663) = (110\ 110\ 011) = D^8 + D^7 + D^5 + D^4 + D + 1$$
$$g_1^{(3)} = (711) = (111\ 001\ 001) = D^8 + D^7 + D^6 + D^3 + 1$$

1. 离散卷积方案

$$c_0 = d * g_1^{(1)} = (111000101101001100000000) * (101101111)$$
$$= (1100\ 0011\ 1000\ 0111\ 0000\ 0001)$$
$$c_1 = d * g_1^{(2)} = (111000101101001100000000) * (110110011)$$
$$= (1000\ 0000\ 1110\ 1010\ 1010\ 0101)$$
$$c_2 = d * g_1^{(3)} = (111000101101001100000000) * (111001001)$$
$$= (1010\ 1100\ 1111\ 1010\ 0000\ 1011)$$

一次将三位编码符号 c_0, c_1, c_2 连接，将得到编码符号序列 c。

$$c = (111\ 100\ 001\ 000\ 001\ 001\ 100\ 100\ 111\ 011\ 011\ 001$$
$$011\ 100\ 111\ 100\ 010\ 000\ 010\ 000\ 001\ 010\ 001\ 111)$$

2. 标量矩阵乘积方案

$$c = d \cdot \boldsymbol{G}$$

这里，$\boldsymbol{G}$ 是一个半无限的生成式矩阵。

$$\boldsymbol{G} = \begin{bmatrix} G_0 & G_1 & G_2 & . & . & . & G_m & & & \\ & G_0 & G_1 & G_2 & . & . & . & G_m & & \\ & & G_0 & G_1 & G_2 & . & . & . & G_m & \\ & & & \vdots & & & & & \vdots & \\ & & & & \vdots & & & & & \vdots \end{bmatrix}$$

矩阵元素 G_i，$0 \leqslant i \leqslant 8$，可以容易地从 $g_1^{(1)}$，$g_1^{(2)}$ 和 $g_1^{(3)}$ 获得，如下：

$$G_0 = 111,\ G_1 = 011,\ G_2 = 101, \cdots,\ G_7 = 110,\ G_8 = 111$$

这样，编码符号码字 c 计算如下：

$$c = d \cdot \boldsymbol{G} = (111000101101001100000000)$$
$$= (111\ 100\ 001\ 000\ 001\ 001\ 100\ 100\ 111\ 011\ 011\ 001\ 011\ 100\ 111\ 100\ 010\ 000\ 010$$
$$000\ 001\ 010\ 001\ 111)$$

$$
\begin{bmatrix}
111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & & & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & & & & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & & & & & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & & & & & & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & & & & & & & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & & & & & & & & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & & & & & & & & & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & & & & & & & & & & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & & & & & & & & & & & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & & & & & & & & & & & & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & & & & & & & & & & & & & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & & & & & & & & & & & & & & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & & & & & & & & & & & & & & & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & & & & & & & & & & & & & & & & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111 \\
 & & & & & & & & & & & & & & & & & & & & & & & 111 & 011 & 101 & 110 & 010 & 101 & 100 & 110 & 111
\end{bmatrix}
$$

这就是期望的编码符号码字 c。

3. 多项式矩阵方案

信息多项式表示为：

$$d(D)=D^{23}+D^{22}+D^{21}+D^{17}+D^{15}+D^{14}+D^{12}+D^{9}+D^{8}$$

生成式多项式矩阵表示为：

$$\boldsymbol{G}(D)=[g_1^{(1)}(D),g_1^{(2)}(D),g_1^{(3)}(D)]$$
$$=[D^8+D^6+D^5+D^3+D^2+D+1,D^8+D^7+D^5+D^4+D+1,D^8+D^7+D^6+D^3+1]$$

模 2 加法器处的编码符号多项式矩阵为：

$$C(D)=d(D)\boldsymbol{G}(D)=[c_0(D),c_1(D),c_2(D)]$$
$$=[D^{31}+D^{30}+D^{25}+D^{24}+D^{23}+D^{18}+D^{17}+D^{16}+D^{8},D^{31}+D^{23}+D^{22}+D^{21}+D^{19}+D^{17}+D^{15}+D^{13}+D^{10}+D^{8},D^{31}+D^{29}+D^{27}+D^{26}+D^{23}+D^{22}+D^{21}+D^{20}+D^{19}+D^{17}+D^{11}+D^{9}+D^{8}]$$

最后，转换开关的时延操作将产生如下的编码输出序列：

$$C(D)=D^2c_0(D^3)+Dc_1(D^3)+c_2(D^3)$$
$$=D^2(D^{93}+D^{90}+D^{75}+D^{72}+D^{69}+D^{54}+D^{51}+D^{48}+D^{24})+D(D^{93}+D^{69}+D^{66}+D^{63}+D^{57}+D^{51}+D^{45}+D^{39}+D^{30}+D^{24})+(D^{93}+D^{87}+D^{81}+D^{78}+D^{69}+D^{66}+D^{63}+D^{60}+D^{57}+D^{51}+D^{33}+D^{27}+D^{24})$$
$$=D^{95}+D^{94}+D^{93}+D^{92}+D^{87}+D^{81}+D^{78}+D^{77}+D^{74}+D^{71}+D^{70}+D^{69}+D^{67}+D^{66}+D^{64}+D^{63}+D^{60}+D^{58}+D^{57}+D^{56}+D^{53}+D^{52}+D^{51}+D^{50}+D^{46}+D^{40}+D^{33}+D^{31}+D^{27}+D^{26}+D^{25}+D^{24}$$

=(111 100 001 000 001 001 100 100 111 011 011 001
011 100 111 100 010 000 010 000 001 010 001 111)

这证明以上三种不同方法可得到相同的输出符号序列(Ⅲ)。

例 3.16 4 800 b/s 帧包含 96 比特，这 96 比特由 80 位比特信息加上 8 位 CRC 比特和 8 位尾比特组成。如例 3.12 所示，(3，1，8)卷积编码器输入计算得到序列(Ⅰ)。

利用图 3.14，模 2 加法器处的编码器输出符号计算如下：

c_0 =(1001010001 1000100001 0111110000 1001100111 1001100011
1111110111 1011100111 0011001010 1100010011 011011)

c_1 =(1110010000 1100100001 0001000000 0001101100 1010101100
0101110010 1101010001 0101010001 0100101001 010111)

c_2 =(1101010001 1001111110 0110100011 0100111110 1100110100)
0010000111 0101000101 1001100011 0001111010 110101

这样，利用图 3.4 的开关可以计算得到如下的连续编码器输出序列：

111011010101000111000000000101 111010000001111001001001001110 000101101110101100000000001001 100001000110111001011111101100 111001010100111001010011100100 100110101110110110000101111101 110011100111100010000101100111 001010100110001010100000101011 100110000001011101011000101110 00111110001110111	(Ⅳ)

例 3.17　9 600 b/s 帧包含 192 比特(20 ms)。参阅例 3.13,传输速率为 9 600 b/s 的 RTC 帧序列计算为序列(II),作为(3, 1, 8)卷积编码器输入。利用图 3.4,三个模 2 加法器处的编码器输出符号分别为:

c_0 =(1110011111 0011001010 0000101001 1000011111 1001100101
0000101001 0001110110 0001100101 0000101001 0111100111
0011001100 1010000100 1010101011 1001111100 1110100111
0100110111 0011001010 0011001000 1001000000 01)

c_1 =(1011100001 1111011011 1110101111 1011110100 1010101000
1110101111 0101001011 0010101000 1110101111 0001010001
0101010101 0001110100 1110000000 1100100101 0110110001
0111000001 0101010001 1110110001 1100101001 01)

c_2 =(1001100000 0010111000 1101010010 1001110110 1100110001
1101010010 0100000000 0100110001 1101010010 0001000101
1001100110 0011101011 0001100001 0000110110 0101001101
1110110101 1001100011 1001111001 1000000000 11)

最后,通过图 3.4 开关按列对 c_0,c_1,c_2 交织,可得到连续编码器输出如下:

111100110011011100100100100110 010010111110001011111000110010
011011010001110001110010011110 111000010011011111100111101100
111001010100111001010100000101 011011010001110001110010011110
000011000110100100010100110010 000001010100111001010100000101
011011010001110001110010011110 000100100111100010000101100111
001010100111001010100111001010 100000101011011010001110001001
110010110001101000100000100101 110010000100111101100111001010
100111110001110010001101100111 001111011010101101000101100111
001010100111001010100000101011 011010110101011011101000000011
1110100001000100000010000001010 001111

(V)

3.2.5　编码符号重复

当数据率低于 9 600 b/s 时,卷积编码器输出符号必须重复,然后再交织。RTC 符号速率随着数据率变化。数据率为 9 600 b/s 时不需要符号重复。对于 4 800 b/s 数据率,每个符号重复 3 次。对于 1 200 b/s,每个编码符号重复 7 次。

对所有数据率,每个编码符号重复按照如下方式进行:

数据率	重复次数/符号	连续发生次数/符号
9 600	0	1
4 800	1	2
2 400	3	4
1 200	7	8

重复后的编码符号将输入块交织器，由于发送占空比不同，因此除了一次编码符号重复以外，其余将全部删除，然后再发送。

例 3.18 重新考虑发送速率为 1 200 b/s 的 RTC 帧。

参阅例 3.14，卷积编码器符号输出序列为

1111000010 0000100110 0100111011 0110010111 0011110001 0000010000 0010100011 11	(Ⅲ)

数据率为 1 200 b/s 的编码符号将重复 7 次(每个符号连续出现 8 次)。因此，重复的符号表示如下：

```
11111111 11111111 11111111 11111111 00000000 00000000 00000000
00000000 11111111 00000000 00000000 00000000 00000000 00000000
11111111 00000000 00000000 11111111 11111111 00000000 00000000
11111111 00000000 00000000 11111111 11111111 11111111 00000000
11111111 11111111 00000000 11111111 11111111 00000000 00000000
11111111 00000000 11111111 11111111 11111111 00000000 00000000
11111111 11111111 11111111 11111111 00000000 00000000 00000000
11111111 00000000 00000000 00000000 00000000 00000000 11111111
00000000 00000000 00000000 00000000 00000000 00000000 11111111
00000000 11111111 00000000 00000000 00000000 11111111 11111111
11111111 11111111
```

这些重复编码符号将输入块交织器，但是它们将不会多次发送，相反，在实际发送之前，除了一次，其余的重复将全部删除。

3.2.6 块交织

长度为 20 ms 的块交织器对反向业务信道上所有编码符号交织，然后再发送。

交织器形成一个 32 行、18 列共 576 个元素的阵列。

低于 9 600 b/s 数据率的重复编码符号按列写入交织器并填满 32 × 18 矩阵。

表 3.2、表 3.17、表 3.18 和表 3.19 表明了发送速率分别为 4 800 b/s，9 600 b/s，2 400 b/s 和 1 200 b/s 时重复编码符号写入交织阵列的操作顺序。

RTC 编码符号将按行从交织器输出。根据发送速率，交织器行将以下列顺序输出。

对 9 600 b/s 数据率(表 3.17)：

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

交织器输出序列以规则的顺序按行发送：

对 4 800 b/s 数据率(表 3.2)：

1 3 2 4 5 7 6 8 9 11 10 12 13 15 14 16

17 19 18 20 21 23 22 24 25 27 26 28 29 31 30 32

交织器输出序列以下列行顺序发送：

$$1+4i,3+4i,2+4i,4+4i,0\leqslant i\leqslant 7$$

对 2 400 b/s 数据率（表 3.18）：

1	5	2	6	3	7	4	8	9	13	10	14	11	15	12	16
17	21	18	22	19	23	20	24	25	29	26	30	27	31	28	32

交织器输出序列以下列行顺序发送：

$$1+8i,5+8i,2+8i,6+8i,3+8i,7+8i,4+8i,8+8i,0\leqslant i\leqslant 3$$

对 1 200 b/s 数据率（表 3.19）：

1	9	2	10	3	11	4	12	5	13	6	14	7	15	8	16
17	25	18	26	19	27	20	28	21	29	22	30	23	31	24	32

交织器输出序列以下列行顺序发送：

$$1+16i,9+16i,2+16i,10+16i,3+16i,11+16i,4+16i,12+16i,5+16i,$$
$$13+16i,6+16i,14+16i,7+16i,15+16i,8+16i,16+16i,i=0,1$$

表 3.17　反向业务信道交织器存储（**9 600 b/s**）（**TIA/EIA/IS-95**）

1	33	65	97	129	161	193	225	257	289	321	353	385	417	449	481	513	545
2	34	66	98	130	162	194	226	258	290	322	354	386	418	450	482	514	546
3	35	67	99	131	163	195	227	259	291	323	355	387	419	451	483	515	547
4	36	68	100	132	164	196	228	260	292	324	356	388	420	452	484	516	548
5	37	69	101	133	165	197	229	261	293	325	357	389	421	453	485	517	549
6	38	70	102	134	166	198	230	262	294	326	358	390	422	454	486	518	550
7	39	71	103	135	167	199	231	263	295	327	359	391	423	455	487	519	551
8	40	72	104	136	168	200	232	264	296	328	360	392	424	456	488	520	552
9	41	73	105	137	169	201	233	265	297	329	361	393	425	457	489	521	553
10	42	74	106	138	170	202	234	266	298	330	362	394	426	458	490	522	554
11	43	75	107	139	171	203	235	267	299	331	363	395	427	459	491	523	555
12	44	76	108	140	172	204	236	268	300	332	364	396	428	460	492	524	556
13	45	77	109	141	173	205	237	269	301	333	365	397	429	461	493	525	557
14	46	78	110	142	174	206	238	270	302	334	366	398	430	462	494	526	558
15	47	79	111	143	175	207	239	271	303	335	367	399	431	463	495	527	559
16	48	80	112	144	176	208	240	272	304	336	368	400	432	464	496	528	560
17	49	81	113	145	177	209	241	273	305	337	369	401	433	465	497	529	561
18	50	82	114	146	178	210	242	274	306	338	370	402	434	466	498	530	562
19	51	83	115	147	179	211	243	275	307	339	371	403	435	467	499	531	563
20	52	84	116	148	180	212	244	276	308	340	372	404	436	468	500	532	564
21	53	85	117	149	181	213	245	277	309	341	373	405	437	469	501	533	565
22	54	86	118	150	182	214	246	278	310	342	374	406	438	470	502	534	566
23	55	87	119	151	183	215	247	279	311	343	375	407	439	471	503	535	567
24	56	88	120	152	184	216	248	280	312	344	376	408	440	472	504	536	568
25	57	89	121	153	185	217	249	281	313	345	377	409	441	473	505	537	569
26	58	90	122	154	186	218	250	282	314	346	378	410	442	474	506	538	570

（续表）

27	59	91	123	155	187	219	251	283	315	347	379	411	443	475	507	539	571
28	60	92	124	156	188	220	252	284	316	348	380	412	444	476	508	540	572
29	61	93	125	157	189	221	253	285	317	349	381	413	445	477	509	541	573
30	62	94	126	158	190	222	254	286	318	350	382	414	446	478	510	542	574
31	63	95	127	159	191	223	255	287	319	351	383	415	447	479	511	543	575
32	64	96	128	160	192	224	256	288	320	352	384	416	448	480	512	544	576

表 3.18 反向业务信道交织器存储（**2 400 b/s**）（**TIA/EIA/IS-95**）

1	9	17	25	33	41	49	57	65	73	81	89	97	105	113	121	129	137
1	9	17	25	33	41	49	57	65	73	81	89	97	105	113	121	129	137
1	9	17	25	33	41	49	57	65	73	81	89	97	105	113	121	129	137
1	9	17	25	33	41	49	57	65	73	81	89	97	105	113	121	129	137
2	10	18	26	34	42	50	58	66	74	82	90	98	106	114	122	130	138
2	10	18	26	34	42	50	58	66	74	82	90	98	106	114	122	130	138
2	10	18	26	34	42	50	58	66	74	82	90	98	106	114	122	130	138
2	10	18	26	34	42	50	58	66	74	82	90	98	106	114	122	130	138
3	11	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139
3	11	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139
3	11	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139
3	11	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139
4	12	20	28	36	44	52	60	68	76	84	92	100	108	116	124	132	140
4	12	20	28	36	44	52	60	68	76	84	92	100	108	116	124	132	140
4	12	20	28	36	44	52	60	68	76	84	92	100	108	116	124	132	140
4	12	20	28	36	44	52	60	68	76	84	92	100	108	116	124	132	140
5	13	21	29	37	45	53	61	69	77	85	93	101	109	117	125	133	141
5	13	21	29	37	45	53	61	69	77	85	93	101	109	117	125	133	141
5	13	21	29	37	45	53	61	69	77	85	93	101	109	117	125	133	141
5	13	21	29	37	45	53	61	69	77	85	93	101	109	117	125	133	141
6	14	22	30	38	46	54	62	70	78	86	94	102	110	118	126	134	142
6	14	22	30	38	46	54	62	70	78	86	94	102	110	118	126	134	142
6	14	22	30	38	46	54	62	70	78	86	94	102	110	118	126	134	142
6	14	22	30	38	46	54	62	70	78	86	94	102	110	118	126	134	142
7	15	23	31	39	47	55	63	71	79	87	95	103	111	119	127	135	143
7	15	23	31	39	47	55	63	71	79	87	95	103	111	119	127	135	143
7	15	23	31	39	47	55	63	71	79	87	95	103	111	119	127	135	143
7	15	23	31	39	47	55	63	71	79	87	95	103	111	119	127	135	143
8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128	136	144
8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128	136	144
8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128	136	144
8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128	136	144

表 3.19　反向业务信道交织器存储(1 200 b/s)(TIA/EIA/IS-95)

1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	69
1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	69
1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	69
1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	69
1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	69
1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	69
1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	69
1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	69
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70
3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63	67	71
3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63	67	71
3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63	67	71
3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63	67	71
3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63	67	71
3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63	67	71
3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63	67	71
3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63	67	71
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72

例 3.18　对 1 200 b/s 数据率，利用表 3.19 的行计算交织器输出。利用编码器输出序列(III)和顺序编号，可获得表 3.20 的交织器输出。

表 3.20　对应发送速率1 200 b/s的交织器输出符号

行编号	交织器输出	行编号	交织器输出
[1]	101000111001000011	[17]	100110100110000101
[9]	100011110101100001	[25]	100000011110010001
[2]	101000111001000011	[18]	100110100110000101

（续表）

行编号	交织器输出	行编号	交织器输出
[10]	100011110101100001	[26]	100000011110010001
[3]	101000111001000011	[19]	100110100110000101
[11]	100011110101100001	[27]	100000011110010001
[4]	101000111001000011	[20]	100110100110000101
[12]	100011110101100001	[28]	100000011110010001
[5]	101000111001000011	[21]	100110100110000101
[13]	100011110101100001	[29]	100000011110010001
[6]	101000111001000011	[22]	100110100110000101
[14]	100011110101100001	[30]	100000011110010001
[7]	101000111001000011	[23]	100110100110000101
[15]	100011110101100001	[31]	100000011110010001
[8]	101000111001000011	[24]	100110100110000101
[16]	100011110101100001	[32]	100000011110010001

例 3.19 对 9 600 b/s 数据率，利用表 3.17 的行计算交织器输出。由于卷积编码器输出表示序列（V），交织器输出根据表 3.21 顺序编号计算。

表 3.21 对应发送速率为**9 600 b/s**的交织器输出符号

行编号	交织器输出	行编号	交织器输出
[1]	101000110001111011	[17]	010101100110011110
[2]	101111001010001000	[18]	010101011000011000
[3]	110001001010010110	[19]	111101010110100110
[4]	101001111110100000	[20]	000010000001101000
[5]	010110000001011110	[21]	001100011011110000
[6]	010110010100101011	[22]	101110010110100000
[7]	110010001110101100	[23]	011000000101000001
[8]	111101110001111010	[24]	011010111110101000
[9]	011101000011010000	[25]	101111001010110101
[10]	001111101101011111	[26]	000101101000011000
[11]	100100001110111100	[27]	011110001111001110
[12]	100110001100010110	[28]	101010110000110010
[13]	010101101001001010	[29]	101000010100110111
[14]	101010110000000000	[30]	010111010000101111
[15]	111000000101001110	[31]	010001101110001011
[16]	111101010001110011	[32]	100100100010011101

例 3.20 对 4 800 b/s 数据率，利用编码符号序列（IV）和类似于表 3.2 的表 3.22 编号行计算交织器输出。

表 3.22　对应4 800 b/s发送速率的交织器输出符号

行编号	交织器输出	行编号	交织器输出
[1]	111000010111111101	[17]	000110110001010011
[3]	110010011001010011	[19]	101101011101100001
[2]	111000010111111101	[18]	000110110001010011
[4]	110010011001010011	[20]	101101011101100001
[5]	101110010010001001	[21]	001111010101001011
[7]	000000100111000111	[23]	111010101010110111
[6]	101110010010001001	[22]	001111010101001011
[8]	000000100111000111	[24]	111010101010110111
[9]	100010111001111111	[25]	001000000100111010
[11]	100110011011000000	[27]	010000110011010101
[10]	100010111001111111	[26]	001000000100111010
[12]	100110011011000000	[28]	010000110011010101
[13]	000011101110010000	[29]	010000101111001101
[15]	100000100000000000	[31]	111101111111101101
[14]	000011101110010000	[30]	010000101111001101
[16]	100000100000000000	[32]	111101111111101101

3.2.7 RTC 正交调制

RTC 调制是 64 阶正交调制。每 6 个编码符号发送使用 Walsh 函数产生的 64 个可能的调制符号之一。表 2.4 给出了这些编号从 0 到 63 的调制符号。由于块交织器输出编码符号速率为 28.8 ks/s，所以正交调制器输出符号速率变为 4.8 ks/s 或 307.2 kc/s。

调制符号可根据下列调制符号指数（MSI）来选择：

$$\mathrm{MSI} = c_0 + 2c_1 + 4c_2 + 8c_3 + 16c_4 + 32c_5$$

其中 c_i 表示二进制编码符号。

通过下列例子来考虑 64 阶正交调制器的输入/输出关系。

例 3.21　发送速率为 9 600 b/s 时，交织器第一行输出为 101000110001111011（参见例 3.19）。

由于 6 个输入符号为 101000，$c_0 = 1$，$c_1 = 0$，$c_2 = 1$，$c_3 = 0$，$c_4 = 0$，$c_5 = 0$，所以调制符号指数 MSI = 5。利用表 2.4，对应 MSI = 5 的调制符号计算为：

01011010010110100101101001011010
01011010010110100101101001011010

类似地，6 个输入符号为 110001 时，有 MSI = 35。对应 MSI = 35 的调制符号为：

01100110011001100110011001100110
10011001100110011001100110011001

6 个输入符号为 111011 时，有 MSI = 55。对应 MSI = 55 的调制符号为：

01101001011010011001011010010110
10010110100101100110100101101001

这样就计算了对应于 18 个编码符号的 3 个 192 位码片调制符号。该输出对应于 32 行中的一

行。

例 3.22 发送速率为 1 200 b/s 时，交织器行以下列顺序输出：

1 9 2 10 3 11 4 12 5 13 6 14 7 15 8 16

17 25 18 26 19 27 20 28 21 29 22 30 23 31 24 32

出于演示的目的，考虑头 2 行，即 1 和 9。交织器对应于这 2 行的输出在例 3.18 已计算出。

第 1 行：101000111001000011

第 9 行：100011110101100001

[计算第 1 行]

1. 6 个输入符号为 101000($c_0=1, c_1=0, c_2=1, c_3=0, c_4=0, c_5=0$)，MSI = 5，利用表 2.4，调制符号为：

0101101001011010010110100101 1010

0101101001011010010110100101 1010

2. 6 个输入符号为 111001($c_0=1, c_1=1, c_2=1, c_3=0, c_4=0, c_5=1$)，MSI = 39，调制符号为：

01101001011010010110100101101001

10010110100101101001011010010110

3. 6 个输入符号为 000011($c_0=0, c_1=0, c_2=0, c_3=0, c_4=1, c_5=1$)，MSI = 48，相应的调制符号为：

00000000000000001111111111111111

11111111111111110000000000000000

[计算第 9 行]

4. 输入符号为 100011，MSI = 49

调制符号 = 01010101010101011010101010101010

10101010101010100101010101010101

5. 输入符号为 110101，MSI = 43

调制符号 = 01100110100110010110011010011001

10011001011001101001100101100110

6. 输入符号为 100001，MSI = 33

调制符号 = 01010101010101010101010101010101

10101010101010101010101010101010

根据输入/输出计算结果，调制符号可归纳如下：

输入(关于第 1 行和第 9 行的交织器输出符号)

101000111001000011
100011110101100001

(Ⅵ)

输出(对应第 1 行和第 9 行的调制符号)

```
01011010010110100101101001011010
01011010010110100101101001011010
01101001011010010110100101101001
10010110100101101001011010010110
00000000000000001111111111111111
11111111111111110000000000000000
01010101010101011010101010101010
10101010101010100101010101010101
01100110100110010110011010011001
10011001011001101001100101100110
01010101010101010101010101010101
10101010101010101010101010101010
```

3.2.8　数据脉冲随机发生器

RTC 交织器输出由时间滤波器选通，允许发送和删除某些交织器输出符号。如图 3.28 所示，发送选通过程随着数据率的不同而变化。对 9 600 b/s 数据率，允许选通所有交织器输出符号发送。对 4 800 b/s 数据率，允许选通一半（50%）的交织器输出符号发送。对2 400 b/s数据率，允许选通四分之一（25%）的交织器输出符号发送。对 1 200 b/s 数据率，允许选通 12.5% 的交织器输出符号发送。

将 20 ms 帧分成 16 个标号为 0 到 15 的功率控制组进行选通操作，如图 3.28 所示。每个功率控制组（PCG）处理 1.25 ms 长的数据。选通的开关操作称为数据脉冲随机发生。数据脉冲随机发生器确保每个符号重复过程的输入编码符号仅发送一次。

选通打开功率组在帧内的位置是伪随机化的。移动台只在选通打开的期间以正常控制功率发送数据。在选通关闭期间，无论哪种功率较大，移动台将相对于最近功率控制组或相对于发射机噪声功率的平均输出功率减少至少 20 dB。

数据脉冲随机发生器产生二进制掩码，可以随机掩蔽掉码重复产生的多余数据。掩码数据由帧数据率和来自于长码的 14 位比特确定。这 14 位 PN 序列是前面一个功率控制组（PCG14）到前一帧的最后一个功率控制组（PCG15）进行扩展时用到的长码的最后 14 比特。

假设这 14 位比特表示为：$\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9, \alpha_{10}, \alpha_{11}, \alpha_{12}, \alpha_{13}$，其中 α_0 表示最旧比特位，α_{13}表示最新比特位。这 14 比特在每个 RTC 帧边界前产生一个功率控制组（1.25 ms）。

数据脉冲随机发生器算法根据选定的不同数据率在标号为 0 到 15 的 PCG 中发送。随机发生算法如下：

数据率	编号的 **PCG**
9 600	0，1，2，3，4，5，6，7，8，9，10，11，12，13，14，15
4 800	α_0，$2+\alpha_1$，$4+\alpha_2$，$6+\alpha_3$，$8+\alpha_4$，$10+\alpha_5$，$12+\alpha_6$，$14+\alpha_7$
2 400	如果 $\alpha_8=0$，则为 α_0；如果 $\alpha_8=1$，则为 $2+\alpha_1$

（续表）

数　据　率	编号的 PCG
	如果 $\alpha_9=0$，则为 $4+\alpha_2$；如果 $\alpha_9=1$，则为 $6+\alpha_3$
	如果 $\alpha_{10}=0$，则为 $8+\alpha_4$；如果 $\alpha_{10}=1$，则为 $10+\alpha_5$
	如果 $\alpha_{11}=0$，则为 $12+\alpha_6$；如果 $\alpha_{11}=1$，则为 $14+\alpha_7$
1 200	如果 $\alpha_8=0$ 且 $\alpha_{12}=0$，则为 α_0；如果 $\alpha_8=1$ 且 $\alpha_{12}=0$，则为 $2+\alpha_1$
	如果 $\alpha_9=0$ 且 $\alpha_{12}=1$，则为 $4+\alpha_2$；如果 $\alpha_9=1$ 且 $\alpha_{12}=1$，则为 $6+\alpha_3$
	如果 $\alpha_{10}=0$ 且 $\alpha_{13}=0$，则为 $8+\alpha_4$；如果 $\alpha_{10}=1$ 且 $\alpha_{13}=0$，则为 $10+\alpha_5$
	如果 $\alpha_{11}=0$ 且 $\alpha_{13}=1$，则为 $12+\alpha_6$；如果 $\alpha_{11}=1$ 且 $\alpha_{13}=1$，则为 $14+\alpha_8$

20 ms 帧均匀分成 16 个标号为 0 到 15 的功率控制组，但是选通打开功率组会针对不同的发送速率而有所区别。基于表 3.23 的图 3.28 是 RTC 变速率的示例。

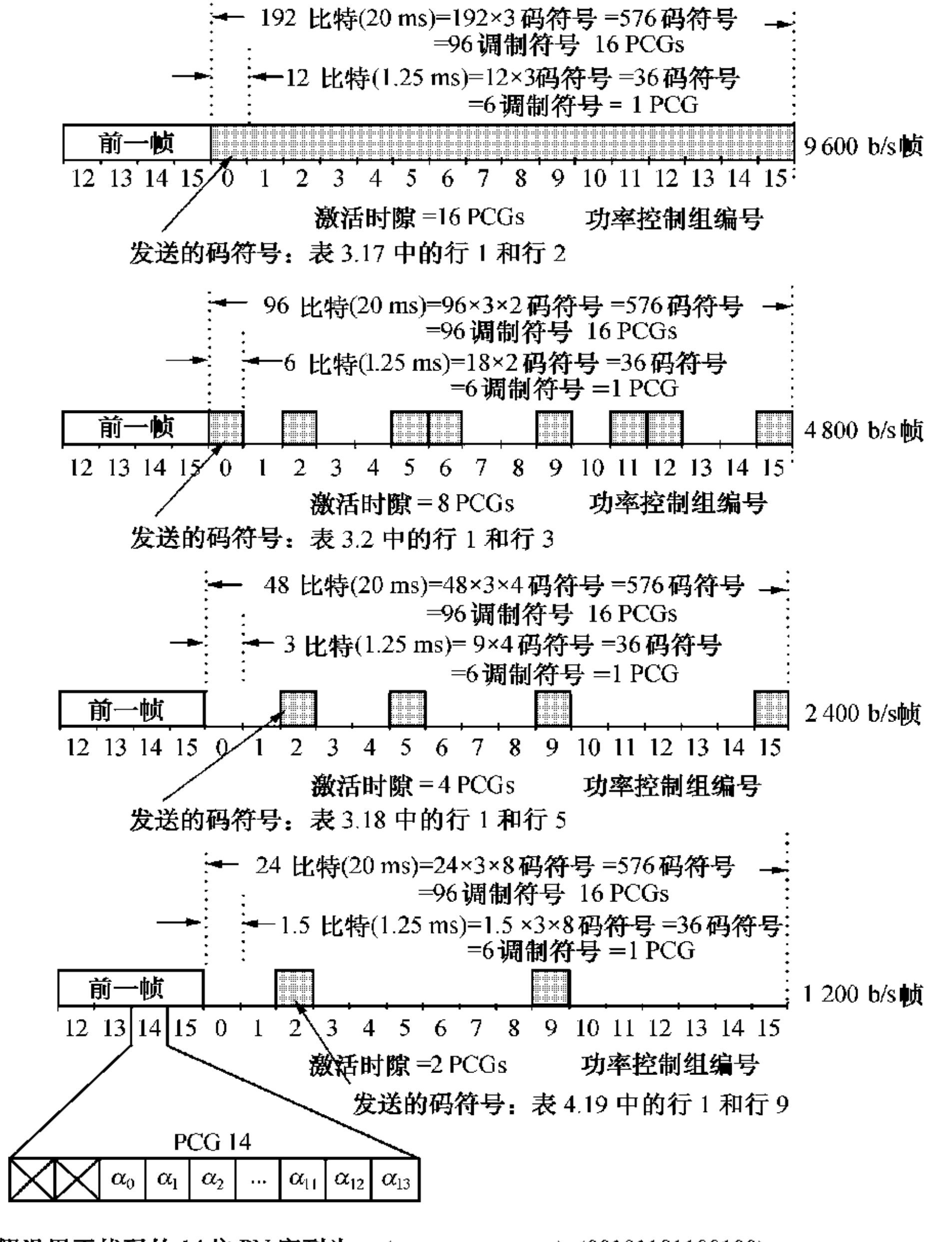


图 3.28　随发送速率变化的数据脉冲随机发生器脉冲训练序列

表 3.23　**RTC** 变速率发送

数据率	每帧信息比特	编码器输出符号	连续发生次数	交织后码符号	调制符号	**PCG**	选通功率组
9 600	192(20ms)	576	1	576	96	16	16(100%)
	12(1.25ms)	36		36	6	1	1 PCG
4 800	96(20ms)	288	2	576	96	16	8(50%)
	6(1.25ms)	18		36	6	1	1 PCG
2 400	48(20ms)	144	4	576	96	16	4(25%)
	3(1.25ms)	9		36	6	1	1 PCG
1 200	24(20ms)	72	8	576	96	16	2(12.5%)
	1.5(1.25ms)	4.5		36	6	1	1 PCG

例 3.23　一帧(20 ms)包括 16 个时隙(PCGs)。

对于传输速率 1 200 b/s,其信息比特等于 24 比特每帧,它将作为卷积编码器输入。由于码率为 1/3,编码器输出的编码符号为 72 比特。经过重复过程得到 576(=72×8)个编码符号。这些编码符号先交织,然后进行调制和发送。因此,调制符号为 96(=576/6)。

对于单个 PCG,持续时间 = 1.25(=20/16)ms,信息比特 = 1.5(=24/16)比特。编码器编码符号 = 4.5(=1.5×3)比特,重复符号 = 36(=4.5×8)比特。Walsh 调制 = 6(=36/6)调制符号。

假设扰码 PN 序列为$(\alpha_0,\ldots,\alpha_{13})$ = (00101101100100),其中 $\alpha_0=0,\alpha_1=0,\alpha_2=1,\alpha_3=0,\alpha_4=1,\alpha_5=1,\alpha_6=0,\alpha_7=1,\alpha_8=1,\alpha_9=0,\alpha_{10}=0,\alpha_{11}=1,\alpha_{12}=0,\alpha_{13}=0$。

这个扰码数据表示了 14 位 PN 序列。

功率控制组编号在 1 200 b/s 数据脉冲随机发生器算法之后确定如下:

如果 $\alpha_8=0$ 和 $\alpha_{12}=0$,则 $\alpha_0=0$,对应编号 0 的 PCG。

如果 $\alpha_{10}=0$ 和 $\alpha_{13}=0$,则 $\alpha_4+8=9$,对应编号 9 的 PCG。

从表 3.20 可以明显地看出第 1 行和第 9 行对应的交织器输出符号代表编号 0 的 PCG,第 18 行和第 26 行对应的交织器输出符号代表编号 9 的 PCG。这样编号 0 的 PCG 编码符号为 101000111001000011100011110101100001,编号 9 的 PCG 编码符号为 100110100110000101100000011110010001。实际上,对应编号 0 的 PCG 编码符号序列为符号序列(VI),对应符号序列(VI)的调制符号表示符号序列(VII)。这样,编号 0 的 PCG(或时隙 0)包含 36 位编码符号

101000 111001 000011 100011 110101 100001　　(VI)

和 6 个调制符号(6 × 64 = 384 码片)。

```
0101101001011010010110100101101001011010010110100101101001011010
0110100101101001011010010110100110010110100101101001011010010110
0000000000000000111111111111111111111111111111110000000000000000
0101010101010101101010101010101010101010101010100101010101010101   (VII)
0110011010011001011001101001100110011001011001101001100101100110
0101010101010101010101010101010110101010101010101010101010101010
```

下一步,考虑编号 9 的 PCG。每 6 位编码符号对应 64 码片调制符号,详见下表。

100110 100110 000101 100000 011110 010001　　(Ⅷ)

```
0101010110101010101010100101010101010101101010101010101001010101
0101010110101010101010100101010101010101101010101010101001010101
0000000011111111000000001111111111111111000000001111111100000000
0101010101010101010101010101010101010101010101010101010101010101
0011110011000011110000110011110000111100110000111100001100111100
0011001100110011001100110011001100111100110011001100110011001100
```
(Ⅸ)

传输速率为 1 200 b/s 时，调制符号序列(Ⅷ)和(Ⅸ)只是分别表示编号 0 和编号 9 的数据脉冲随机发生器输出流。

3.2.9　直接序列扩展

考虑反向业务信道的长码直接序列扩展。对 RTC，扩展操作包括数据脉冲随机发生器输出流和长码的模 2 加。

长码周期为 $2^{42}-1$ 个码片，由码生成式的 LFSR 抽头多项式 $p(x)$ 指定：

$$p(x)=1+x+x^2+x^3+x^5+x^6+x^7+x^{10}+x^{16}+x^{17}+x^{18}+x^{19}+x^{21}+x^{22}+x^{25}+x^{26}+x^{27}+x^{31}+x^{33}+x^{35}+x^{42} \tag{3.17}$$

长码的每个 PN 码片由输入的 42 位掩码和序列生成式的 42 级 LFSR“与”操作得到的结果进行模 2 求和而产生的，如图 3.8 所示。长码提供了有限的保密性。根据移动台传输的信道类型的不同，长码掩码也不一样。长码掩码由 42 位二进制序列组成，它能惟一标识长码。

在 TRC 发送时，移动台使用 2 个长码掩码中的 1 个，即对移动台 ESN 来说是惟一的公用长码掩码或私有长码掩码。公用长码掩码按如下方式产生：

$M_{41}\sim M_{32}$：设置为 1100011000。

$M_{31}\sim M_0$：设置为移动台的 ESN 比特的某种排列。

移动台的 ESN 比特排列定义为：

$$\text{ESN}=(E_{31},E_{30},E_{29},E_{28},E_{27},E_{26},\cdots,E_3,E_2,E_1,E_0),$$

ESN 比特排列必须指定为一种能降低长码之间相关性的方式，如下所示：

$$\text{排列后的 ESN}=(E_0,E_{31},E_{22},E_{13},E_4,E_{26},E_{17},E_8,E_{30},E_{21},E_{12},E_3,E_{25},E_{16},E_7,E_{29},E_{20},E_{11},E_2,E_{24},E_{15},E_6,E_{28},E_{19},E_{10},E_1,E_{23},E_{14},E_5,E_{27},E_{18},E_9)$$

公用长码掩码格式如图 3.29。

41　　32	31　　0
1100011000	**排列后的**ESN=($E_0,E_{31},E_{22},\cdots,E_{27},E_{18},E_9$)
MSB	LSB

图 3.29　公用长码掩码格式

私有长码掩码现在用不着，因为对外的使用分发由 TIA 控制。

例 3.24　再次考虑发送速率 1 200 b/s 的情况。

如果移动台的 ESN 比特位为 01001100101100111010011101100111，则排列后的 ESN 变为

10010111111000100101010110011101101。公用长码掩码格式（参见 3.29）表示为 11000110001001011111100010010101100111011101（由最高有效位到最低有效位）。如果公用长码掩码以相反的顺序安排，则变为 10110111001101010010001111101001000110001 1（由最低有效位到最高有效位）。参见图 3.8，反向的 42 位长码掩码和 LFSR 的 42 级输出都将输入"与"门。"与"门输出进行模 2 加。如果模 2 加法器输出的整数为偶数，则输出二进制值 0，如果是奇数，则输出二进制值 1。

通过式(3.17)，生成序列可以表示为 111101110010000011110110011100010101000000。假设 LFSR 的初始内容为 100000000000000000000000000000000000000000。

将这些序列运用到图 3.8，会产生如下的长码 PN 码片：

1011 0111 0011 0101 0010 0011 1110 1001 0001 1000 1111 1011 1010 1000 0010 0011 1000 0001 1000 1100 0000 0001 0100 0100 0011	(X)

该序列表示第一组 100 次移位操作产生的部分长码。

最后，考虑正交调制符号和长码 PN 码片"异或"操作进行的直接序列扩展(DSS)。对 1 200 b/s数据率，调制符号表示数据脉冲随机发生器输出流。为简便起见，只考虑长码序列(X)的第 1 行。

由于 PN 码片/Walsh 码片等于 4，所以计算方法如下：

调制符号：	0	1	0	1	1	0	1	0	0	0
长码码片：	1011	0111	0011	0101	0010	0011	1110	1001	0001	1000
异或(⊕)	1011	1000	0011	1010	1101	0011	0001	1001	0001	0111

这样，长码 PN 码片产生的 DSS 计算如下：

1011 1000 0011 1010 1101 0011 0001 1001 0001 011	(XI)

例 3.25　传输速率为 4 800 b/s 时，交织器输出将导致表 3.2 中的下列行顺序：1,3,2,4,5,7,6,8,9,11,10,12,13,15,14,16,17,19,18,20,21,23,22,24,25,27,26,28,29,31,30,32。交织器输出符号已在表 3.22 中计算好。

现在只考虑最初的第 1 行和第 3 行，它们将输入 64 阶正交调制器。这样，第 1 行和第 3 行交织器输出为：

111000 010111 111101

110010 011001 010011

64 阶正交调制器输出流(或调制符号)使用下表的调制符号指数(MSI)计算。

交织器输出	调制符号索引	调制器输出流(参见表 2.10)
111000	7	0110100101101001011010010110100 1
		01101001011010010110100101101001
010111	58	00110011110011001100110000110011
		11001100001100110011001111001100
111101	47	01101001100101100110100110010110

（续表）

交织器输出	调制符号系列	调制器输出流（参见表 2.10）
		1001011001101001100101100110100l
110010	19	01100110011001101001100110011001
		01100110011001101001100110011001
011001	38	00111100001111000011110000111100
		11000011110000111100001111000011
010011	50	00110011001100111100110011001100
		11001100110011000011001100110011

该调制器输出流等价于编号 0 的 PCG。

传输速率为 4 800 b/s 时，发送门允许交织器输出符号的一半进行发送。如果用于数据扰码的 14 位 PN 序列假设为：

$(\alpha_0,\alpha_1,\alpha_2,\alpha_3,\alpha_4,\alpha_5,\alpha_6,\alpha_7,\alpha_8,\alpha_9,\alpha_{10},\alpha_{11},\alpha_{12},\alpha_{13})$

$=(00101101100100)$，发送在下列编号的 PCG 进行：

$\alpha_0, 2+\alpha_1, 4+\alpha_2, 6+\alpha_3, 8+\alpha_4, 10+\alpha_5, 12+\alpha_6, 14+\alpha_7$

由于 $\alpha_0 = 0$，则 PCG 编号为 0；　　由于 $2+\alpha_1 = 2$，则 PCG 编号为 2；

由于 $4+\alpha_2 = 5$，则 PCG 编号为 5；　　由于 $6+\alpha_3 = 6$，则 PCG 编号为 6；

由于 $8+\alpha_4 = 9$，则 PCG 编号为 9；　　由于 $10+\alpha_5 = 11$，则 PCG 编号为 11；

由于 $12+\alpha_6 = 12$，则 PCG 编号为 12；　　由于 $14+\alpha_7 = 15$，则 PCG 编号为 15。

这样，可以看到分配的选通的 PCG 在 20 ms 帧内刚好占 1/2，如图 3.30 所示。

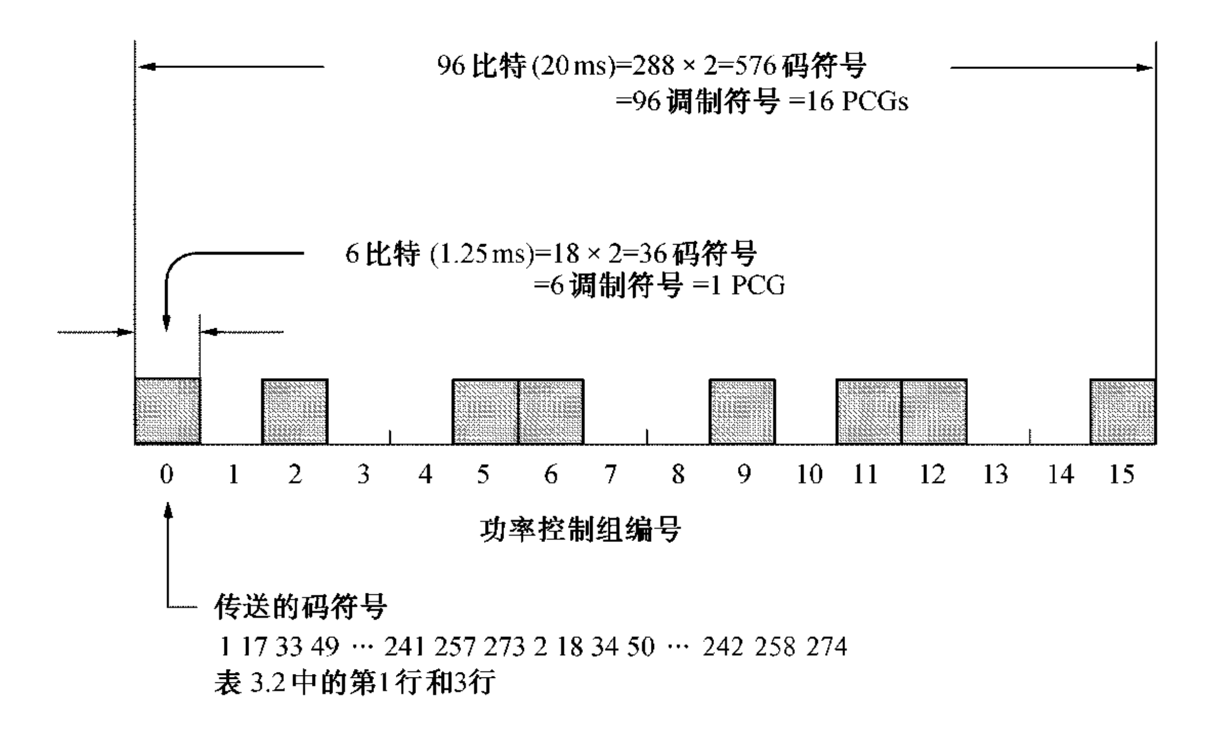


图 3.30　对 RTC 4 800 b/s 帧（20 ms）选通功率控制组的分配（16 个 PCG 的 1/2）

选通的 PCG 实际上是数据脉冲随机发生器输出，含有时隙 0，2，5，6，9，11，12 和 15。

根据表 3.22，编号为 1 和 3 的行对应 PCG 编号 0，编号为 5 和 7 的行对应 PCG 编号 2，编号为 10 和 12 的行对应 PCG 编号 5 等。传输速率为 4 800 b/s 时，RTC 交织器输出流被选通，允许发送

交织器输出符号的一半（即激活时隙）并删除其他符号。表 3.24 显示了对应选通组（PCG）的 36 位交织器输出符号。

表 3.24　对应于激活 PCG 编号的交织器输出符号

行编号	PCG 编号	交织器输出符号（36 比特）
1,3	0	111000010111111101 110010011001010011
5,7	2	101110010010001001 000000100111000111
10,12	5	100010111001111111 100110011011000000
13,15	6	000011101110010000 100000100000000000
18,20	9	000110110001010011 101101011101100001
22,24	11	001111010101001011 111010101010110111
25,27	12	001000000100111010 010000110011010101
30,32	15	010000101111001101 111101111111101101

考虑 PCG 编号 2 和 PCG 编号 5。对应调制器输入流（交织器输出符号）的调制器输出流（调制符号）计算如下：

编号为 6,9,12 和 15 的 PCG 的计算留给读者作为练习。

PCG 编号	调制符号索引	调制符号（数据脉冲随机发生器输出）
	29	0101101010100101101001010101101001011010101001011010010101011010
	18	0011001100110011110011001100110000110011001100111100110011001100
	36	00001111000011110000111100001111111110000111100001111000011110000
2	0	0000000000000000000000000000000000000000000000000000000000000000
	57	0101010110101010101010100101010110101010010101010101010110101010
	56	0000000011111111111111110000000011111111000000000000000011111111
	17	0101010101010101101010101010101001010101010101011010101010101010
	39	0110100101101001011010010110100110010110100101101001011010010110
	63	0110100110010110100101100110100110010110011010010110100110010110
5	25	0101010110101010101010100101010101010101101010101010101001010101
	54	0011110000111100110000111100001111000011110000110011110000111100
	0	0000000000000000000000000000000000000000000000000000000000000000

3.2.10　正交相位扩展

直接序列扩展之后，RTC 将进行正交扩展。I 和 Q 路导频 PN 序列分别基于下列 LFSR 抽头多项式产生，这些 PN 序列周期为 2^{15} 个码片。这些多项式与用于接入信道的多项式相同：

$$P_I(x) = 1 + x^5 + x^7 + x^8 + x^9 + x^{13} + x^{15}$$

$$P_I = (1000010111000101) \qquad \text{（矢量形式）}$$

$$P_Q(x) = 1 + x^3 + x^4 + x^5 + x^6 + x^{10} + x^{11} + x^{12} + x^{15}$$

$$P_Q = (1001111000111001) \qquad \text{（矢量形式）}$$

正交扩展序列 I 和 Q 很容易分别由 P_I 和 P_Q 导频序列和 DSS 流进行"异或"运算产生。

I 信道流 I 和 Q 信道流 Q 是 I 和 Q 导频码片和 DSS 流进行"异或"运算得到的输出序列。

首先考虑导频 PN 码的产生。利用给定的抽头序列 P_I 和 P_Q，LFSR 初始内容为

(100000000000000)时,可以计算 I 和 Q 导频 PN 码片,如表 3.4 所示。

同相和正交相位导频 PN 序列分别为:

↓

$\bar{P}_I$ = 00000000000000010101001001110100011011110011001000
0011110000100011010010101101010101011101001001111

↓

P_Q = 00000000000000010011110101110101101001110001010011
1001110001101100011101001100010010010001100110001111

请注意:在 $\bar{P}_I$ 和 $\bar{P}_Q$ 连续输出 14 个 0 之后再插入 1 个二进制值 0(参见向下箭头)。

例 3.26 利用 P_I 和 P_Q 导频 PN 序列,I 信道流 I 由 P_I 和 DSS 流进行"异或"计算得到,Q 信道流 Q 由 P_Q 和 DSS 流进行"异或"计算得到。

发送速率为 1 200 b/s 时,同相信道序列计算为:

DSS: 10111000 00111010 11010011 00011001 00010111

⊕

$\bar{P}_I$: 00000000 00000001 01010010 01110100 01101111

I: 10111000 00111011 10000001 01101101 01111000

而正交相位序列计算为:

DSS: 10111000 00111010 11010011 00011001 00010111

⊕

$\bar{P}_Q$: 00000000 00000001 00111101 01110101 10100111

Q: 10111000 00111011 11101110 01101100 10110000

例 3.27 发送速率为 4 800 b/s 时,DSS 流可以通过对数据脉冲随机发生器输出(PCG)和长码序列(X)进行"异或"计算产生。发送速率为 1 200 b/s 时,长码 PN 码片与发送速率为4 800 b/s 时相同,即

1011 0111 0011 0101 0010 0011 1110 1001 0001 1000

这只表示整个长码 PN 码片的开始部分。

关于 MSI = 7,对应 PCG 编号 0 的数据脉冲随机发生器输出为

01101001011101001011010010110100 1
01101001011101001011010010110100 1

考虑最开始 10 个符号 0110100101,它对应下列 40 个码片 000011111111000011110000000 0011110000 1111,因为 PN 码片/Walsh 码片比率为 4。

现在可以按如下方式计算 DSS 流:

随机发生器输出(PCG 编号 0):	0	1	1	0	1	0	0	1	0	1
⊕										
长码码片	1011	0111	0011	0101	0010	0011	1110	1001	0001	1000
DSS 流:	1011	1000	1100	0101	1101	0011	1110	0110	0001	0111

数据率为 4 800 b/s 或 1 200 b/s 时，I 信道导频 PN 码片 $\bar{P}_I$ 和 Q 信道导频 PN 码片 $\bar{P}_Q$ 保持不变，它们分别为：

$\bar{P}_I$：00000000 00000001 01010010 01110100 01101111

$\bar{P}_Q$：00000000 00000001 00111101 01110101 10100111

这样，同相序列 I 和正交相位序列 Q 可以分别通过对 DSS 序列和 $\bar{P}_I$ 或 $\bar{P}_Q$ 进行"异或"计算产生。

同相序列 I 按如下方法获得：

DSS 流：	10111000	11000101	11010011	11100110	00010111
$\oplus$					
$\bar{P}_I$：	00000000	00000001	01010010	01110100	01101111
同相 I：	10111000	11000100	10000001	10010010	01111000

正交相位序列 Q 如下获得：

DSS 流：	10111000	11000101	11010011	11100110	00010111
$\oplus$					
$\bar{P}_Q$：	00000000	00000001	00111101	01110101	10100111
同相 Q：	10111000	11000100	11101110	10010011	10110000

3.2.11　RTC 正交相移键控（QPSK）

前面章节得到的 I 和 Q 数据流将输入到 I 和 Q 基带滤波器。由 Q 信道导频 PN 码片扩展的数据流 Q 延时了半个码片时间。令 $I(t)$ 和 $Q(t)$ 为基带滤波器输出流，由于 Q 延时半个码片时间，脉冲流 $I(t)$ 和 $Q(t)$ 定时偏移 $T_b/2$ 秒。

将每个 $I(t)$ 和 $Q(t)$ 幅度调制到载波的余弦和正弦函数上去，可以得到偏移 QPSK 波形 $s(t)$，如图 3.16 所示。同相流 $I(t)$ 以 +1 或 −1 对余弦函数进行幅度调制，产生一个 BPSK 波形。正交相位流 $Q(t)$ 对正弦函数进行调制，产生另一个与前面余弦函数产生波形正交的 BPSK 波形。这样，这两个正交 BPSK 波形的和就产生 QPSK 波形。Q 延时半个码片时间 $T_b/2$ 秒得到的偏移 QPSK 脉冲流在下面的例子中给出。

例 3.28　画出发送速率为 1 200 b/s 的偏移 QPSK 脉冲流。正如前面计算的，同相信道序列 I 为：

$$I = (10111000\ 00111011\ \underline{10000001}\ 01101101\ 01111000)$$

正交相位信道序列 Q 为：

$$Q = (10111000\ 00111011\ \underline{11101110}\ 01101100\ 100110000)$$

出于演示目的，分别从 I 流和 Q 流选取第三组 8 位元，I_3 = (10000001) 和 Q_3 = (11101110)。Q_3 延时半个码片时间 $T_b/2$ 秒得到的偏移 QPSK 脉冲流在下面的图 3.31 给出。

例 3.29　计算发送速率为 2 400 b/s 时的偏移 QPSK 脉冲流。

同相序列 I 和正交相位序列 Q 已经在例 3.27 计算出，它们分别为：

$$I = (10111000\ 11000100\ 10000001\ \underline{10010010}\ 01111000)$$

$$Q = (10111000\ 11000100\ 11101110\ \underline{10010011}\ 101100000)$$

考虑从 I 流和 Q 流选取第 4 组 8 位元，I_4 = (10010010) 和 Q_4 = (10010011)。Q_4 延时半个码

片时间 $T_b/2$ 秒得到的偏移 QPSK 脉冲流在下面的图 3.32 给出。

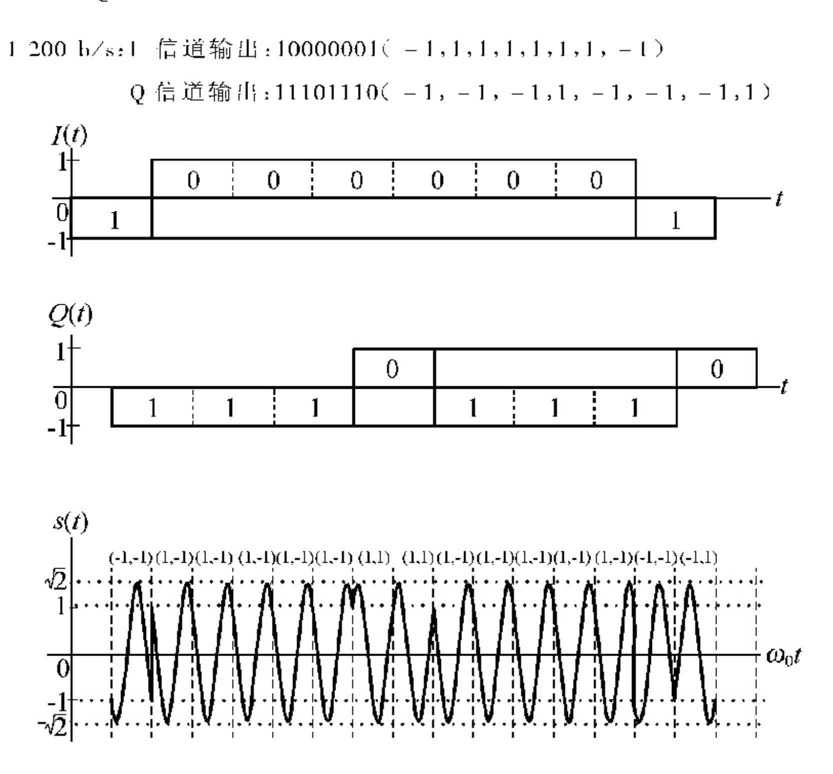
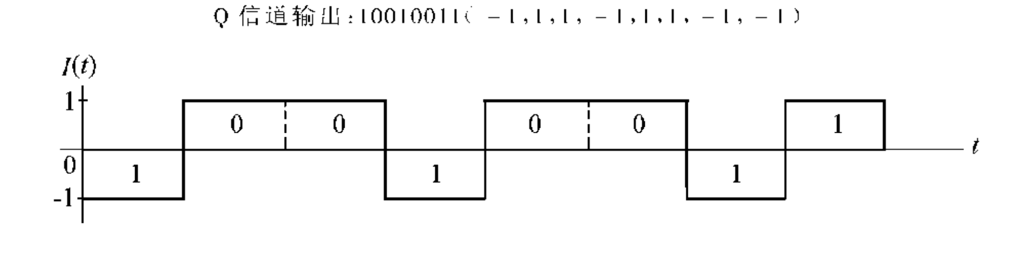
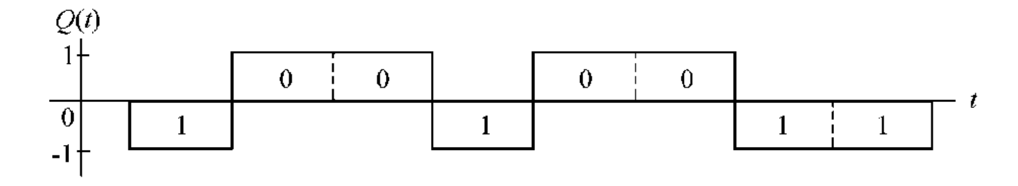


图 3.31 以 1 200 b/s 速率发送对应 I_3 和 Q_3 序列的偏移 QPSK 波形

4 800 b/s: I 信道输出:10010010(−1,1,1, −1,1,1, −1,1)

Q 信道输出:10010011(−1,1,1, −1,1,1, −1, −1)





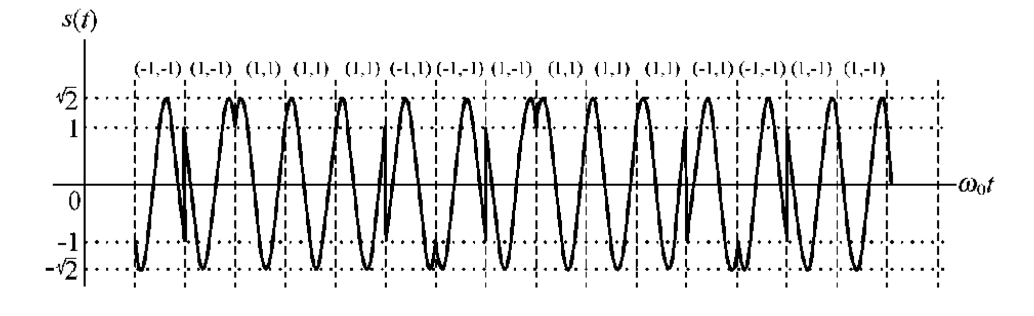


图 3.32 以 4 800 b/s 速率发送对应 I_4 和 Q_4 序列的偏移 QPSK 波形

例 3.30 发送速率为 9 600 b/s 时,参见表 3.21,表中计算了对应于从 1 到 32 行的整个交织器输出。对应于第 1 行 101000110001111011 的 192 码片调制符号序列在例 3.21 中给出。192 码片调

制符号只表示下列正交调制器输出。

01011010010110100101101001011010 01011010010110100101101001011010
01100110011001100110011001100110 10011001100110011001100110011001
01101001011010011001011010010110 10010110100101100110100101101001

由于发送速率为 9 600 b/s 时不需要重复,所以正交调制器输出流和数据脉冲随机发生器是相同的码片流。换句话说,发送速率为 9 600 b/s 时,不需要数据脉冲随机发生器。

由于排列后的 ESN 为 10010111110001001010110011101101,所以 42 位公用长码掩码序列表示为 110001100010010111110001001010110011101101[由最高有效位到最低有效位(MSB-LSB)]。

这样,长码 PN 码片产生如下:

LFSR 初始内容:100000000000000000000000000000000000000000

生成式抽头序列:111101110010000011110110011100010101000000

公用长码掩码比特:110001100010010111110001001010110011101101(LSB-MSB)

利用图 3.8,长码 PN 码片由"与"门输出进行模 2 求和产生。100 次计时得到的 PN 码片为:

1011 0111 0011 0101 0010 0011 1110 1001 0001 1000

1111 1011 1010 1000 0010 0011 1000 0001 1000 1100

0000 0001 0100 0011

只考虑最初的 40 码片,对正交调制器输出和长码 PN 码片进行"异或"操作可得到 DSS 数据序列。

调制符号:	0	1	0	1	1	0	1	0	0	1
⊕										
长码 PN 码片	1011	0111	0011	0101	0010	0011	1110	1001	0001	1000
DSS 序列:	1011	1000	0011	1010	1101	0011	0001	1001	0001	0111

发送速率为 9 600 b/s 时分别得到的 I 信道导频 PN 码片 $\bar{P}_I$ 和 Q 信道导频 PN 码片 $\bar{P}_Q$。

插入 0

↓

$\bar{P}_I$ = 00000000000000010101001001110100011011110011001000

0011110000100011010010101101010101011101001001111

插入 0

↓

$\bar{P}_Q$ = 00000000000000010011110101110101101001110001010011

10011100011011000111010011000100100100011001100011

这样,同相信道流 I 和正交信道流 Q 分别由对 DSS 流和 $\bar{P}_I$ 或 $\bar{P}_Q$ 进行"异或"操作得到,如下所示:

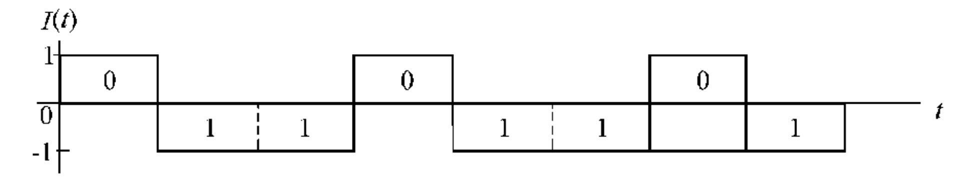
DSS 序列:	10111000	00111010	11010011	00011001	00010111
⊕					
$\bar{P}_I$:	00000000	00000001	01010010	01110100	01101111

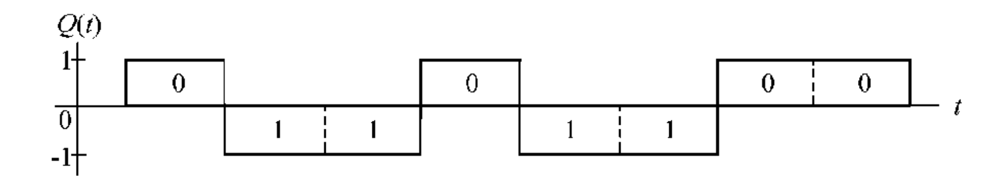
同相 I：	10111000	00111011	10000001	<u>01101101</u>	01111000
DSS 序列：	10111000	00111010	11010011	00011001	00010111
$\oplus$					
P_Q：	00000000	00000001	00111101	01110101	10100111
正交相 Q：	10111000	00111011	11101110	<u>01101100</u>	10110000

考虑从 I 流和 Q 流选取第 4 组 8 位元，I_4 =（01101101）和 Q_4 =（01101100）。Q_4 延时半个码片时间 $T_b/2$ 秒得到的偏移 QPSK 脉冲流在图 3.33 中给出。

9 600 b/s： I 信道输出：01101101（1，－1，－1，1，－1，－1，1，－1）

Q 信道输出：01101100（1，－1，－1，1，－1，－1，1，1）





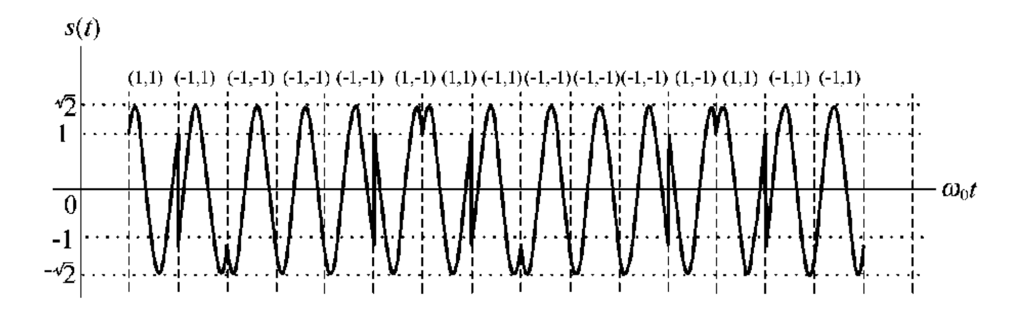


图 3.33 以 9 600 b/s 速率发送的对应 I_4 和 Q_4 序列的偏移 QPSK 波形

第 4 章　前向 CDMA 信道

在 CDMA 指定的频率段利用特定导频 PN 偏移进行传输的一个或多个编码信道组成前向信道。与编码信道相关的是导频信道、同步信道、寻呼信道和前向业务信道。基站利用导频 PN 序列的时间偏移区分前向信道。在一个 CDMA 蜂窝系统中,时间偏移可以复用。

CDMA 前向信道的整体结构如图 4.2、图 4.10 及图 4.26 所示。下面将讨论针对 CDMA 基站操作的要求。

4.1　前向信道

前向信道由如下信道组成:导频信道、至多 1 个同步信道、至多 7 个寻呼信道以及一些前向业务信道。这些编码信道通过适当的 Walsh 函数正交相位扩展,然后由固定码率为 1.228 8 Mc/s 的正交移相导频 PN 序列扩频。在一个基站中通过频分复用的方式可获得多个 CDMA 前向信道。

通过基站传输的信道的典型例子如图 4.1 所示。该例描述了 64 个可用信道:1 个导频信道、1 个同步信道、7 个寻呼信道、55 个业务信道。

编码信道编号为 0(W_0),通常用做导频信道。

如果有同步信道,其编号一般为 32(W_{32})。如果有寻呼信道,其编号通常是从 1 到 7(从 W_1 到 W_7)。余下的编号(W_8,W_{31},W_{32},…,W_{63})用于前向业务信道。

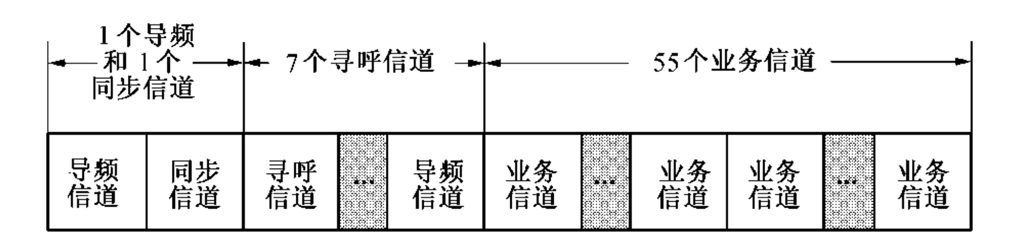


图 4.1　基站传输的前向 CDMA 信道

在 CDMA 前向链路上传输的编码信道通过固定码率为 1.228 8 Mc/s 的 Walsh 函数扩展,以此保证前向 CDMA 信道的所有编码信道正交化。正交相位扩展后,所有编码信道正交相位扩展。扩频之后,I 和 Q 脉冲输入到 I 和 Q 基带滤波器中。基带滤波后,正交相位扩展输出的二进制 I 和 Q 会被映射到指定的相位转换。

4.2　导频信道

导频信道是指 CDMA 前向信道中基站不间断发送的非调制扩频信号,用于发送导频信息的信道。导频信道描述如图 4.2 所示。基站支持所有 CDMA 信道不间断发送导频信号。移动台在所有时间(在时隙模式下不接收除外)检测导频信道。在基站覆盖范围内的移动台利用导频信号来同步。导频信道

是一个参考信道，它允许移动台查询 CDMA 前向信道的信息，从而提取相干载波进行相干解调。在导频和同步过程中，处于初始化状态的移动台查询和同步于基站。

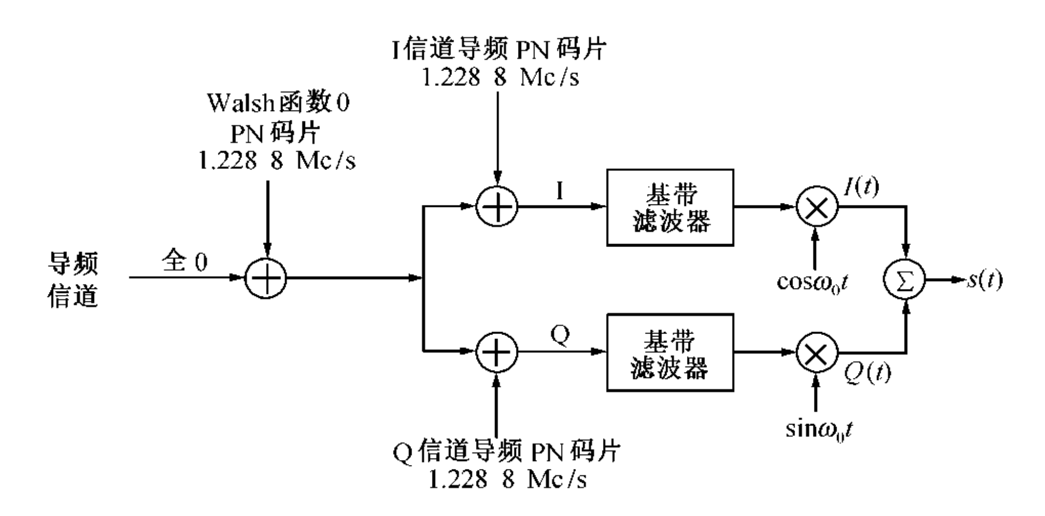


图 4.2 前向 CDMA 信道的导频信道

在导频信道查询子状态，移动台查询选定的 CDMA 系统的导频信道。一旦进入该子状态，移动台调谐于编号等于 CDMACH 的 CDMA 信道上，为导频信道设置编码信道并搜寻导频信道。如果移动台在 T_{20m} = 15 秒内查询导频信道，移动台立即进入同步信道查询子状态。如果移动台在 T_{20m} 内不查询导频信道，移动台进入系统探测子状态（将在第 5 章讨论）。T_{20m} 表示保持在初始化状态的导频信道查询子状态的最长时间。

在 CDMA 蜂窝电话系统中，蜂窝在所有链路频率上发送导频载波信号。导频载波信号让移动台进行初始系统同步并且提供来自蜂窝的信号的时间、频率和相位。移动台不停追踪该信号。导频信号的发送功率电平的变化控制着蜂窝覆盖面积。

每个蜂窝发送的导频载波使用同一扩频码，但相位偏移不同。使用同样编码的目的是让移动台通过搜索一个信号的所有相位而获得小区的码相位，从而与系统同步，最强的信号与小区的码相位对应。

4.2.1 Walsh 函数正交相位扩展

从图 4.2 中可以看出导频信号是全零信号，因此，Walsh 函数（或者调制符号序号）是 0，相应地，由 Walsh 函数 0 产生的 Walsh 码片用 0000…000（64 个 0）表示，64 个 Walsh 码作为正交相位扩展的输入。

4.2.2 正交相位扩展

正交扩频后，导频信号要经过如图 4.2 所示的正交相位扩展。称为导频 PN 序列的扩频序列是长为 2^{15} = 32 768 个 PN 码片组成的四相序列。为了获得周期为 2^{15} 的 I 路和 Q 路导频 PN 序列，要在 LFSR 序列 $i(n)$ 和 $q(n)$ 中连续 14 个 0 输出后插入 0，如例 4.1 所示。

扩频序列由如下这些线性反馈移位寄存器抽头多项式而来。对于同相序列来说，抽头多项式如下所示：

$$P_I(x) = 1 + x^5 + x^7 + x^8 + x^9 + x^{13} + x^{15} \tag{4.1}$$

或 $$P_I = (1000010111000101)$$

对于正交相位序列，抽头多项式为：

$$P_Q(x) = 1 + x^3 + x^4 + x^5 + x^6 + x^{10} + x^{11} + x^{12} + x^{15} \tag{4.2}$$

或　　$P_Q=(1001111000111001)$

式(4.1)和式(4.2)与式(3.3)和式(3.4)一样。

I 路和 Q 路导频 PN 序列分别由基于抽头多项式 $P_I(x)$和 $P_Q(x)$的最大长度线性反馈移位寄存器产生。

表示 $P_I(x)$和 $P_Q(x)$的 LFSR 如图 4.3 和图 4.4 所示。导频序列的码率是1.228 8 Mc/s。同样的 I 路和 Q 路导频 PN 序列可由互逆多项式 $i(x)=x^nP_I(x^{-1})$和 $q(x)=x^nP_Q(x^{-1})$生成。

$$i(x)=1+x^2+x^6+x^7+x^8+x^{10}+x^{15} \tag{4.3}$$

和

$$q(x)=1+x^3+x^4+x^5+x^9+x^{10}+x^{11}+x^{12}+x^{15} \tag{4.4}$$

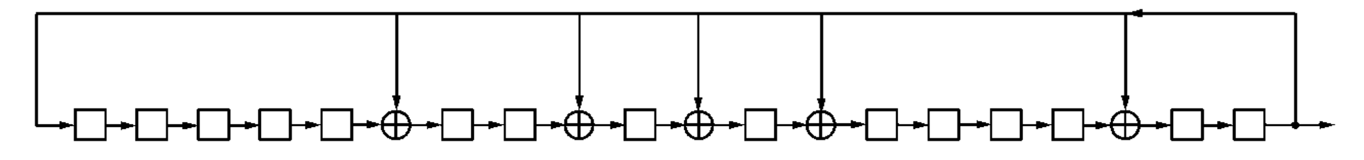


图 4.3　基于抽头多项式

$P_I(x)=1+x^5+x^7+x^8+x^9+x^{13}+x^{15}$的 15 级 LFSR

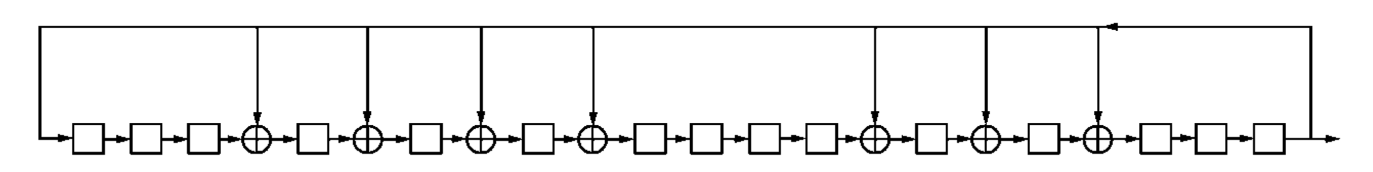


图 4.4　基于抽头多项式

$P_Q(x)=1+x^3+x^4+x^5+x^6+x^{10}+x^{11}+x^{12}+x^{15}$的 15 级 LFSR

利用抽头多项式 $i(x)$和 $q(x)$，基于式(4.3)和式(4.4)的 LFSR 由图 4.5 和图 4.6 描述。

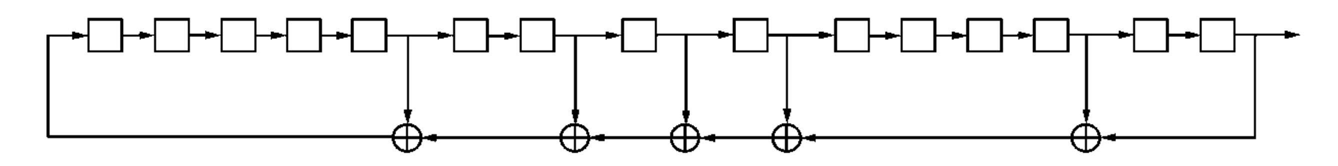


图 4.5　基于互逆抽头多项式

$i(x)=1+x^2+x^6+x^7+x^8+x^{10}+x^{15}$的 15 级 LFSR

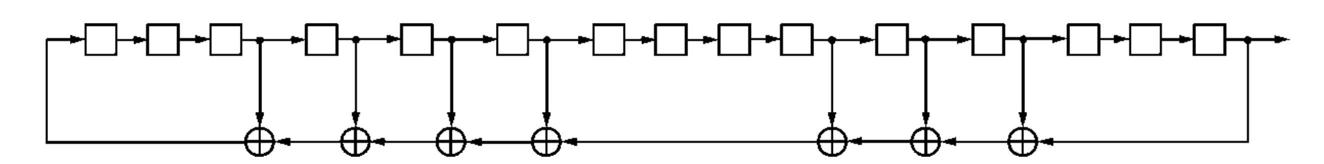


图 4.6　基于互逆抽头多项式

$q(x)=1+x^3+x^4+x^5+x^9+x^{10}+x^{11}+x^{12}+x^{15}$的 15 级 LFSR

称为 LFSR 序列的二进制序列 $i(n)$和 $q(n)$由如下线性操作获得。

$$i(n)=i(n-15)\oplus i(n-10)\oplus i(n-8)\oplus i(n-7)\oplus i(n-6)\oplus i(n-2) \tag{4.5}$$

和

$$q(n)=q(n-15)\oplus q(n-12)\oplus q(n-11)\oplus q(n-10)\oplus q(n-9)\oplus q(n-5)\oplus q(n-4)\oplus q(n-3) \tag{4.6}$$

其中 $i(n)$和 $q(n)$($1\leqslant n\leqslant 32\ 767$)代表二进制值 0 或 1,$i(15)=1$ 和 $q(15)=1$,$\oplus$代表模 2 加。

正交相位扩展输出的二进制系列 I 和 Q 通过 I 路和 Q 路导频序列 $\bar{P}_I$ 和 $\bar{P}_Q$、模 2 加的全零输入序列计算而得。

例 4.1 首先考虑导频 PN 码片的产生。同相抽头序列 P_I = (1000010111000101)和正交相抽头序列 P_Q = (100111000111001)。设 LFSR 的初始内容为(100000000000000),键控序列 $\bar{P}_I$ 和 $\bar{P}_Q$ 的计算过程如下:

插入 0

↓

$\bar{P}_I$ =(00000000000000010101001001110100011011110011001000
001111000010001101001010110101010101110100100111111)

这是同相导频 PN 码片($1\leqslant n\leqslant 99$),其中 n = 键控数。

插入 0

↓

$\bar{P}_Q$ =(0000000000000001001111010111010110100111000101001 1
1001110001101100011101001100010010010001100110001 11)

这是正交相位的导频 PN 码片($1\leqslant n\leqslant 99$),其中 n = 键控数。注意,连续输出 14 个 0 后,在 $\bar{P}_I$ 和 $\bar{P}_Q$ 中箭头所示的地方补加 0。

正交相位扩展输出的 I 路系列 I 和 Q 路系列 Q 通过将 $\bar{P}_I$ 和 $\bar{P}_Q$ 与模 2 加的全零输入序列的异或计算而得。

全 0 输入	00000000 00000000 00000000 00000000 00000000
$\oplus$	
$\bar{P}_I$	00000000 00000001 01010010 01110100 01101111
I	00000000 00000001 01010010 01110100 01101111
全 0 输入	00000000 00000000 00000000 00000000 00000000
$\oplus$	
P_Q	000000000 00000001 00111101 01110101 10100111
Q	000000000 00000001 00111101 01110101 10100111

在导频信道中,可以发现 I 信道流 I 其实是同相导频 PN 序列 P_I,而 Q 信道流 Q 其实是正交相位的导频 PN 序列 $\bar{P}_Q$。

例 4.2 讨论从多项式 $i(x)$和 $q(x)$中获取 $i(n)$和 $q(n)$序列。因为$i(n)=0$($1\leqslant n\leqslant 14$)且 $i(15)=1$,则 $i(n)$在 $16\leqslant n\leqslant 32\ 767$ 范围内的值可以利用 $i(n)=\sum_{k=15,10,8,7,6,2} i(n-k)$,$1\leqslant n\leqslant 32\ 767$ 产生。

$$i(16)=i(1)\oplus i(6)\oplus i(8)\oplus i(9)\oplus i(10)\oplus i(14)$$
$$=0\oplus 0\oplus 0\oplus 0\oplus 0\oplus 0\oplus 0\oplus 0=0,\ n=16 \text{ 时}$$
$$i(17)=i(2)\oplus i(7)\oplus i(9)\oplus i(10)\oplus i(11)\oplus i(15)$$
$$=0\oplus 0\oplus 0\oplus 0\oplus 0\oplus 0\oplus 0\oplus 1=1,\ n=17 \text{ 时}$$
$$i(18)=i(3)\oplus i(8)\oplus i(10)\oplus i(11)\oplus i(12)\oplus i(16)$$

$= 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 = 0$，$n = 18$ 时

$i(19) = i(4) \oplus i(9) \oplus i(11) \oplus i(12) \oplus i(13) \oplus i(17)$

$= 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 = 1$，$n = 19$ 时

.　　　　　　.

.　　　　　　.

.　　　　　　.

因此，同相导频 PN 序列 P_I 通过在 $i(n)$ 中 14 个连续 0 输出位置（如箭头）插入 0 获得，如下所示：

插入 0

↓

$\bar{P}_I$ =（00000000000000010101001001110100011011110011001000
00111100000100011010010101101010101011101001001111 1）

类似地，正交相位的导频 PN 序列 $\bar{P}_Q$ 可利用 $q(n) = 0(1 \leqslant n \leqslant 14)$，$q(15) = 1$ 和 $q(n) = \sum_{k=15,10,8,7,6,2} i(n-k)(1 \leqslant n \leqslant 32\,767)$ 计算。

正交相位的导频 PN 序列通过在 $q(n)$ 中 14 个连续 0 之后（如箭头所示）插入 0 获得。

插入 0

↓

$\bar{P}_Q$ =（00000000000000010011110101110101101001110001010011
100111000110110001110100110001001001000110011000111）

正交相位扩展输出的 I 路序列 I 和 Q 路序列 Q 产生与例 4.1 一样的数据序列。

4.2.3　基带滤波

正交相位扩展后，I 路和 Q 路数据流输入到 I 路和 Q 路基带滤波器中。设 $S(f)$ 是基带滤波器的频谱，滤波器的归一化频率响应限定为：在通带 $0 \leqslant f \leqslant f_p$ 是 $\pm\delta_1$，在阻带 $f \leqslant f_s$ 是 $-\delta_2$，如图 3.15 所示。一组特定参数值是：$\delta_1 = 1.5$ dB，$\delta_2 = 40$ dB，$f_p = 590$ kHz 和 $f_s = 740$ kHz。

设 $S(t)$ 为基带滤波器的冲激响应。则 $S(t)$ 应满足如下等式：

$$\text{均方差} = \sum_{k=0}^{47} [aS(kT_s - \tau) - h(k)]^2 \leqslant 0.03 \tag{4.7}$$

其中 a 和 τ 是最小均方差常数。常数 T_s 等于 $1/4(1.2288 \times 10^6) = 203.451$ ns，刚好等于一个 PN 码片的 1/4 时间。

通过表 4.1，系数 $h(k)$ 值（$20 \leqslant k \leqslant 27$）也可以获得如图 4.7 所示的最小均方误差。

表 4.1　系数 $h(k)$

k	$h(k)$
0,47	−0.025 288 315
1,46	−0.034 167 931
2,45	−0.035 752 323
3,44	−0.016 733 702
4,43	0.021 602 514
5,42	0.064 938 487
6,41	0.091 002 137
7,40	0.081 894 974
8,39	0.037 071 157
9,38	−0.021 998 074
10,37	−0.060 716 277
11,36	−0.051 178 658
12,35	0.007 874 526
13,34	0.084 368 728
14,33	0.126 869 306
15,32	0.094 528 345
16,31	−0.012 839 661
17,30	−0.143 477 028
18,29	−0.211 829 088
19,28	−0.140 513 128
20,27	0.094 601 918
21,26	0.441 387 140
22,25	0.785 875 640
23,24	1.0

（在 TIA/EIA/IS-95 之后）

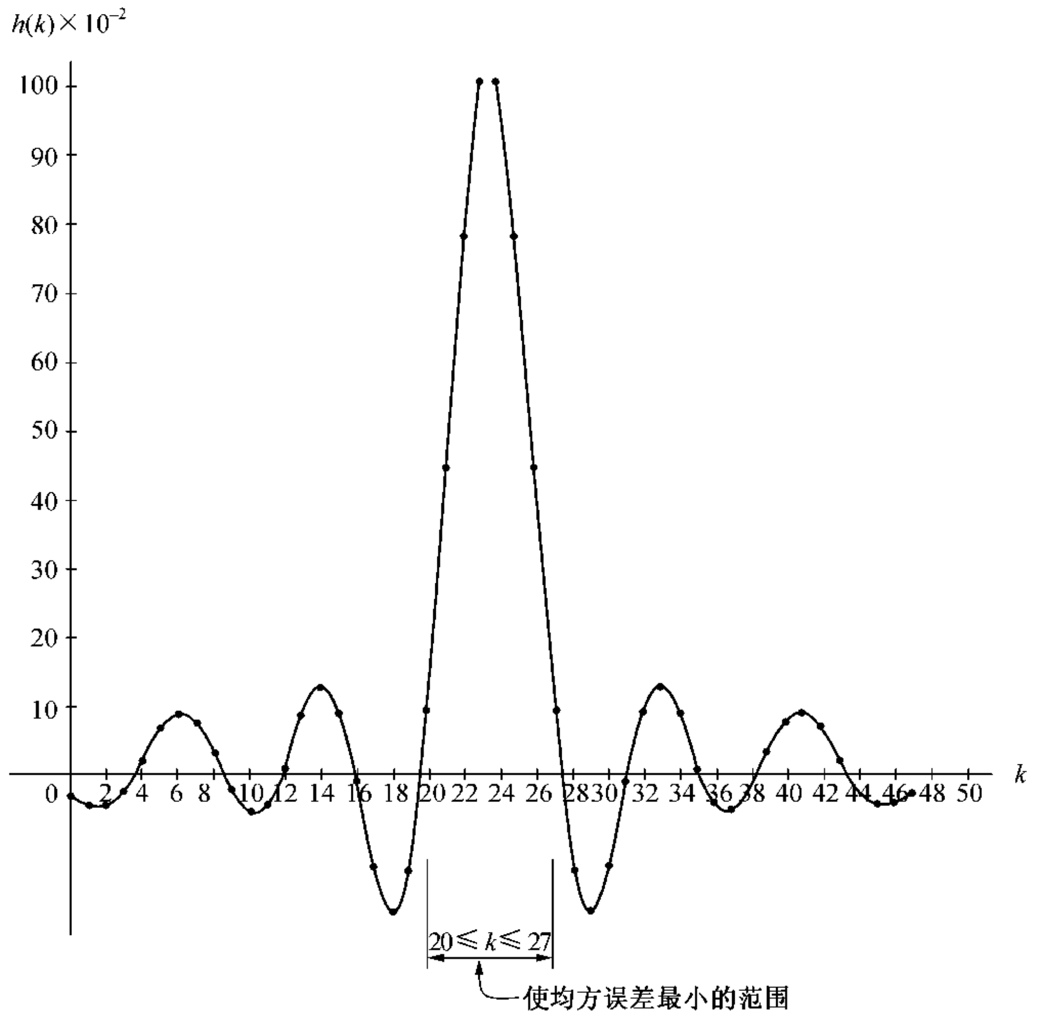


图 4.7　系数 $h(k)$的值（$0\leqslant k\leqslant 47$）

4.2.4　正交相移键控

设 $I(t)$ 和 $Q(t)$ 是基带滤波器的两路输出信号。在非偏移 QPSK 中，两路基带信号步调一致，因此，载波相位每 T_b 秒会改变。

如图 4.2 所示，利用 $I(t)$ 和 $Q(t)$ 对正弦和余弦载波进行幅度调制得到正交 QPSK 波形 $s(t)$。同相信号 $I(t)$ 用幅度 1 或 -1 来对余弦信号进行幅度调制，得到一个 BPSK 波形，而 $Q(t)$ 则对正弦信号进行调制得到正交于余弦函数的另一个 BPSK 波形。因此，这两路正交 BPSK 波形和就形成了 QPSK 波形。

设 $s(t)$ 为 QPSK 波形，表示如下：

$$\begin{aligned} s(t) &= I(t)\cos\omega_0 t + Q(t)\sin\omega_0 t \\ &= \sqrt{2}\cos(\omega_0 t - \theta(t)) \end{aligned} \tag{4.8}$$

其中：

$$I(t)=\sqrt{2}\cos\,\theta(t),\ Q(t)=\sqrt{2}\,\sin\,\theta(t),\ \theta(t)=\arctan\frac{Q(t)}{I(t)}$$

相位 $\theta(t)$ 是相对于周期序列的初始相位偏移。

对特定 $I(t)$，$Q(t)$ 值的 QPSK 信号 $s(t)$，可通过选择 $\theta(t)$ 值来决定，如下所示：

$$\begin{aligned}
&1.\ \theta(t)=\frac{\pi}{4}\text{时},\ I(t)=1,\ Q(t)=1,\ s(t)=\sqrt{2}\cos(\omega_0 t-\frac{\pi}{4})\\
&2.\ \theta(t)=\frac{3\pi}{4}\text{时},\ I(t)=-1,\ Q(t)=1,\ s(t)=\sqrt{2}\cos(\omega_0 t-\frac{3\pi}{4})\\
&3.\ \theta(t)=-\frac{3\pi}{4}\text{时},\ I(t)=-1,\ Q(t)=1,\ s(t)=\sqrt{2}\cos(\omega_0 t+\frac{3\pi}{4})\\
&4.\ \theta(t)=-\frac{\pi}{4}\text{时},\ I(t)=1,\ Q(t)=-1,\ s(t)=\sqrt{2}\cos(\omega_0 t+\frac{\pi}{4})
\end{aligned} \tag{4.9}$$

分析公式(4.9)，与 CDMA 信道(导频、寻呼、前向业务信道)匹配的 $I(t)$，$Q(t)$ 值总结如表 4.2。

表 4.2　前向 CDMA 信道的 $I(t)$，$Q(t)$ 映射

$\theta(t)$	$I(t)$		$Q(t)$	
	NRZ	(二进制值)	NRZ	(二进制值)
$\pi/4$	1	(0)	1	(0)
$3\pi/4$	-1	(1)	1	(0)
$-3\pi/4$	-1	(1)	-1	(1)
$-\pi/4$	1	(0)	-1	(1)

利用表 4.2，信号相位点及其转换关系结果如图 4.8 所示。基带滤波后，正交相位扩展的输出 $I(t)$，$Q(t)$ 符合表 4.2 的相位。

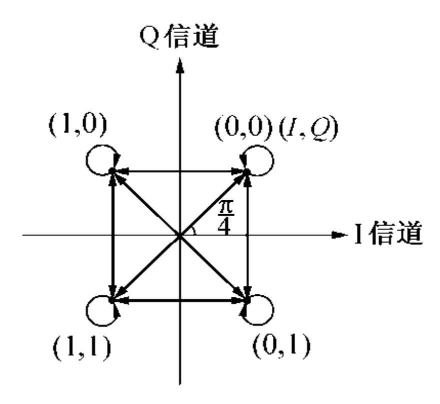


图 4.8　前向 CDMA 信道的信号相位点及其转换关系

表 4.3 **QPSK** 波形 $s(t)$ 的数据图

$\omega_0 t$	$s(t)=\sqrt{2}\cos(\omega_0 t-\theta(t))$			
	$(1,1),\theta(t)=\pi/4$	$(-1,1),\theta(t)=3\pi/4$	$(-1,-1),\theta(t)=-3\pi/4$	$(1,-1),\theta(t)=-\pi/4$
0	1	−1	−1	1
$\pi/4$	$\sqrt{2}$	0	$-\sqrt{2}$	0
$\pi/2$	1	1	−1	−1
$3\pi/4$	0	$\sqrt{2}$	0	$-\sqrt{2}$
π	−1	1	1	−1
$-3\pi/4$	$-\sqrt{2}$	0	$\sqrt{2}$	0
$-\pi/2$	−1	−1	1	1
$-\pi/4$	0	$-\sqrt{2}$	0	$\sqrt{2}$
2π	1	−1	1	1

利用表 4.3，基于 $I(t)$，$Q(t)$ 的 QPSK 脉冲序列可由如下例子画出。

例 4.3 同相序列 I 和正交相位的序列 Q 在例 4.1 中已经计算出，分别是：

I: 00000000	00000001	01010010	01110100	01101111
Q: 00000000	00000001	00111101	01110101	10100111

如果画出在方括号中的 (I,Q) 序列对，则 QPSK 波形 $s(t)$ 如图 4.9 所示。

导频信道

I 信道输出：01010010（1，−1，1，−1，1，1，−1，1）

Q 信道输出：00111101（1，1，−1，−1，−1，−1，1，−1）

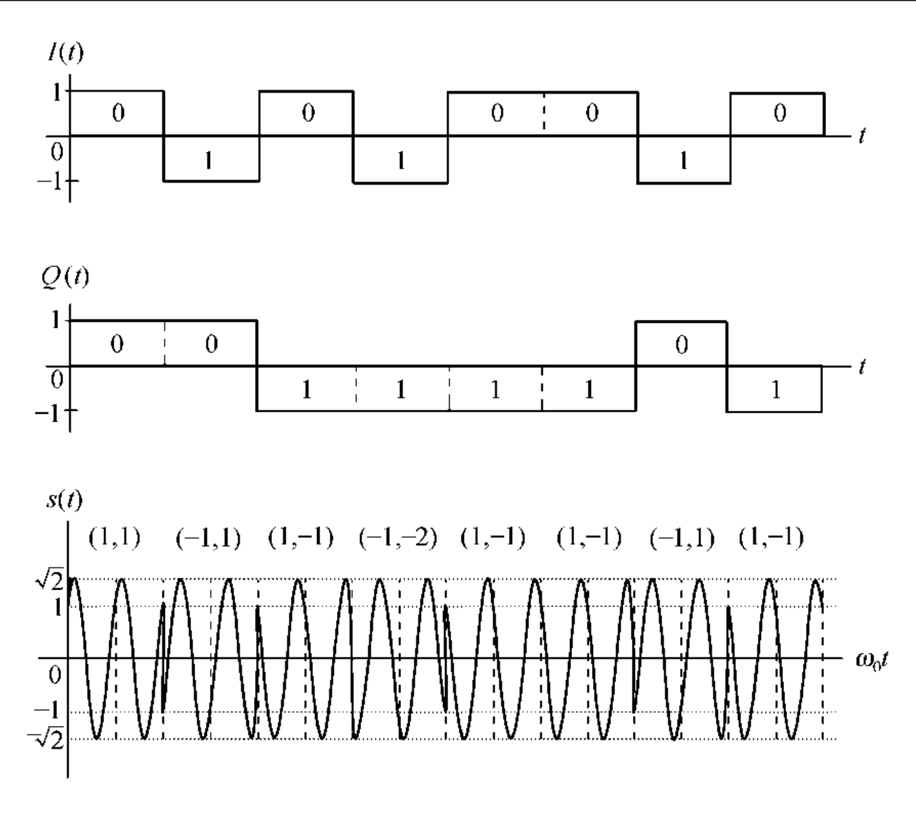


图4.9　导频信道QPSK波形图

4.3　同步信道

CDMA前向信道中编号为32的信道(W_{32})是同步信道,用于向移动台发送同步消息。同步信道速率固定为1 200 b/s。同步信道在发送前要经过卷积编码,而且所有卷积编码符号重复一次。例如,在块交织之前连续出现2次,同步信道中所有符号经过重复后再进行块交织。同步信道使用时间跨度为26.667(=128/4800)ms的分组交织,该跨度与4800 s/s字符速率的128个调制字符相对应。同步信道帧持续时间为26.667 ms。在CDMA前向信道中的同步信道利用码率固定为1.228 8 Mc/s的Walsh函数来扩展,从而使给定CDMA信道中的所有编码信道都正交化。

总之,同步信道上的信号是经过编码、交织、扩频和正交调制等步骤产生的,该信号使移动台获得初始同步时间。同步信道结构(1 200 b/s)如图4.10所示。

用于同步的I路和Q路导频序列把有同样偏移的PN序列当做基站的导频信号。移动台利用同步信息进行同步调整。移动台一旦同步完成,它通常不再接收同步信号,但当设备关机后重新开机时,还需要重新进行同步。一旦移动台查询到系统,除非再次开机,移动台都不会使用同步信道。一旦移动台通过查询导频信道获得导频PN序列的同步,就很容易获得信道同步。这是因为同步信道用相同的导频PN序列来扩频,而且同步信道中的帧结构和交织时间与导频PN序列一致。

由3个同步信道帧构成信道超帧($26.667\times3=80$ ms)。同步信道发送的消息从同步信道超帧的起始点开始,如图4.11所示。

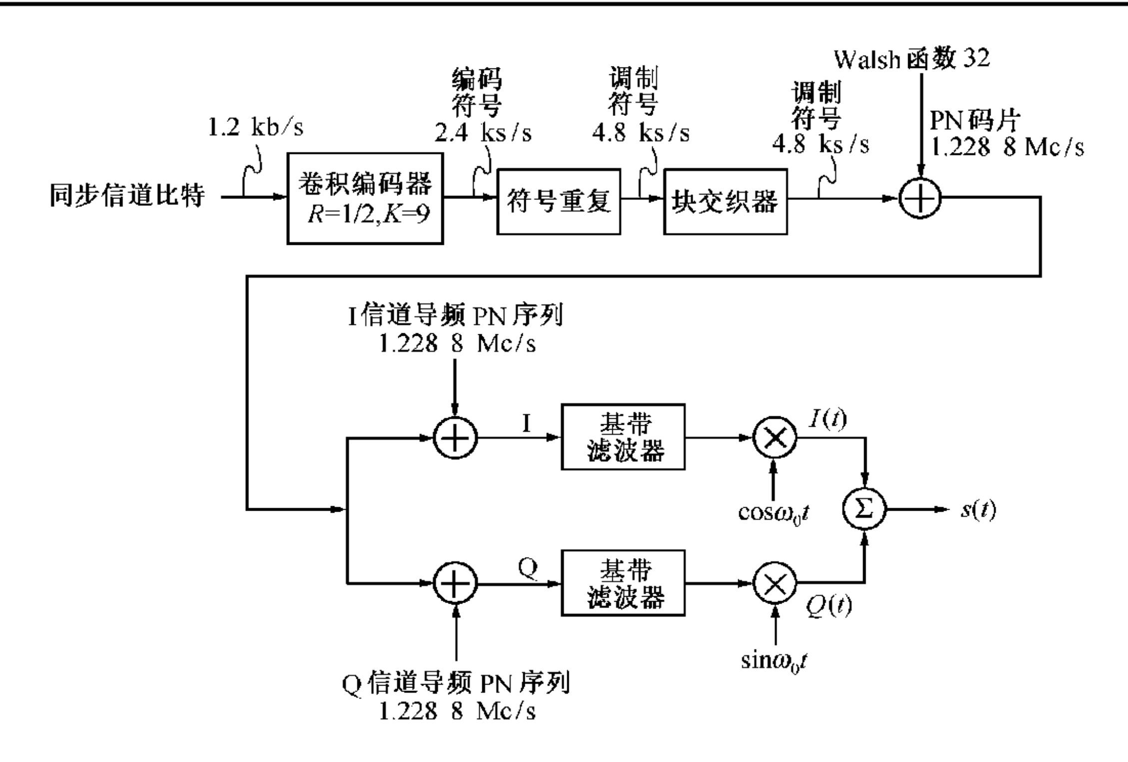


图 4.10 同步信道结构(1 200 b/s)

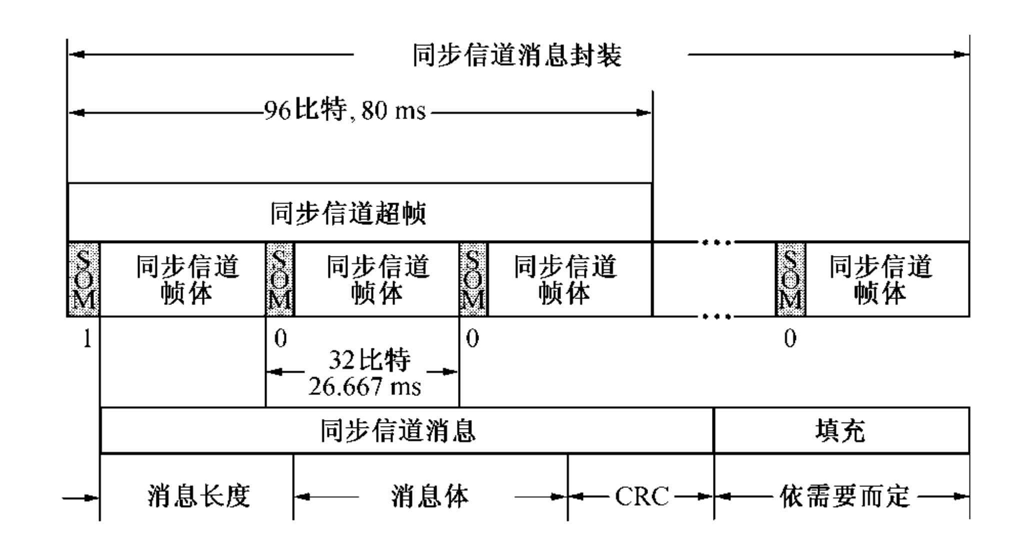


图 4.11 同步信道超帧结构

最后,在表 4.4 中列出同步信道的调制参数。

表 4.4 同步信道调制参数

参　　数	数据速率	单　　位
PN 码片速率	1.228 8	Mc/s
码率	1/2	比特/编码符号
码重复	2	调制符号/编码符号
调制符号速率	4 800	s/s
PN 码片/调制符号	456	PN 码片/调制符号
PN 码片/比特	1 024	PN 码片/比特

4.3.1　同步信道编码

卷积码(n,k,m)表示码率为 $R = k/n$，编码器阶数为 $m = K-1$，其中 K 是约束长度。同步信道用(2,1,8)卷积码来编码，其码率为 $R = 1/2$，约束码长 $K = 9$。卷积码是从延时数据序列中选定的抽头处做模 2 加得到的。因此，数据序列延时长为 $K-1$。编码器生成式序列分别是 $g_1^{(1)} =$ 753(10 进制) = (111101011)(二进制)和 $g_1^{(2)} =$ 561(10 进制) = (101110001)(二进制)。

另一方面，生成式多项式也可写成

$$g_1^{(1)}(x) = 1 + x + x^2 + x^3 + x^5 + x^7 + x^8 \tag{4.10}$$

$$g_1^{(2)}(x) = 1 + x^2 + x^3 + x^4 + x^8 \tag{4.11}$$

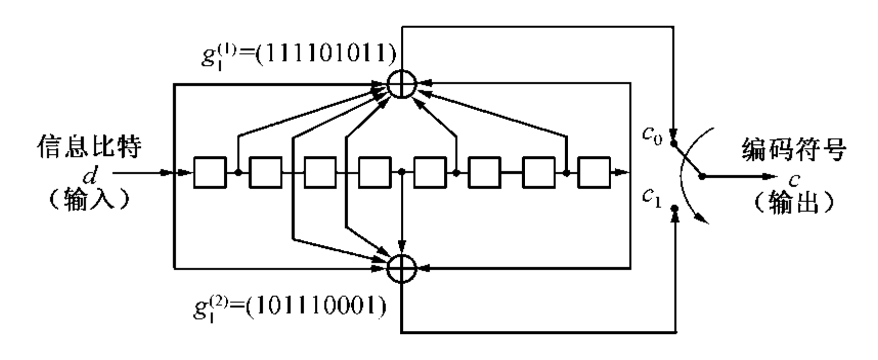
因为 $k = 1, n = 2, m = 8$，该卷积编码器包含一个输入、一个 8 级线性移位寄存器和 2 个模 2 加法器，如图 4.12(a)，(b)所示。因为码率 $R = 1/2$，对每一信息位的输入都会在模 2 加法器输出两个编码符号。其中由生成式序列 $g_1^{(1)}$ 生成的 c_0 是第一个输出，由生成式序列 $g_1^{(2)}$ 生成的 c_1 是第二个输出。卷积编码器的初始状态是全零，初始化后首先输出的符号是由生成式序列 $g_1^{(1)}$ 产生的。

例 4.4　如图 4.12 中卷积编码器所示：两个生成式多项式分别是 $g_1^{(1)} =$ (111101011)和 $g_1^{(2)} =$ (101110001)。对 24 比特信息序列 $d =$ (1001110100 11100010100101)，则编码输出的 32 位比特编码符号 c_0 和 c_1 计算如下。

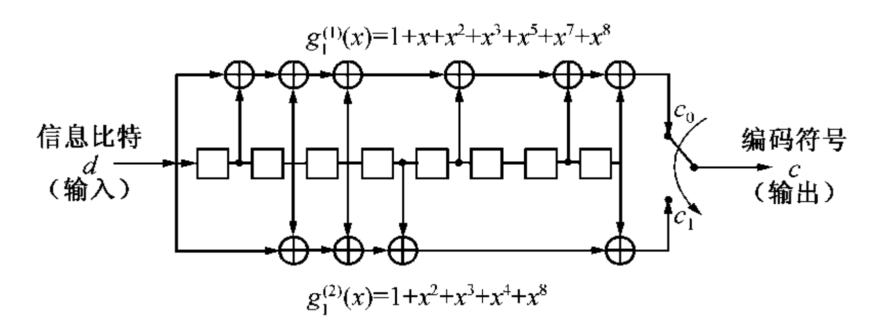
$$c_0 = d_0 + d_1 + d_2 + d_3 + d_5 + d_7 + d_8 \tag{4.12}$$

$$c_1 = d_0 + d_2 + d_3 + d_4 + d_8$$

设卷积编码器的初始内容为 0，由 $g_1^{(1)} = 753$ 生成的 c_0 是第一个输出，由生成序列 $g_1^{(2)} = 561$ 生成的 c_1 是第二个输出。



(a) 基于生成矢量的编码



(b) 基于生成多项式的编码

图 4.12 （2,1,8）卷积编码器

32 位的编码符号序列由 48 位编码符号序列经码率为 1/2 传输和添加 8 位编码而得到。

当信息序列顺序读入编码器中时，c_0 和 c_1 计算如下：

c_0 =（1110001000 0000110100 0110101001 11）

c_1 =（1010000010 0111110010 1010000101 01）

由一对串联复接（c_0，c_1）可以得到单个输出序列，如下所示：

c =（11101100000010000100000101011111
001001000110110010001001001110 11）

下面将演示可替代的几种分析方法。

例 4.5　重新考虑生成式序列为 $g_1^{(1)}$ =（111 101 011）和 $g_1^{(2)}$ =（101 110 001）的（2,1,8）卷积编码。

给定输入信息序列 d =（1001 1101 0011 1000 1010 0101），则输出码序列可计算如下：

1．离散卷积方法

$c_0 = d * g_1^{(1)}$ =（1001 1101 0011 1000 1010 0101）*（111101011）
=（1110 0010 0000 0011 0100 0110 1010 0111）

$c_1 = d * g_1^{(2)}$ =（1001 1101 0011 1000 1010 0101）*（101110001）
=（1010 0000 1001 1111 0010 1010 0001 0101）

经过一对串联复接（c_0，c_1），编码符号 c 如下：

c =（11 10 11 00 00 00 10 00 01 00 00 01 01 01 11 11
01 10 01 00 01 10 11 00 10 00 10 01 00 11 10 11）

2．分级矩阵方法

$$c = d \cdot \boldsymbol{G}$$

其中 $\boldsymbol{G}$ 是半无穷生成矩阵。

$$\boldsymbol{G} = \begin{bmatrix} G_0 G_1 G_2 \cdots G_m & & & \\ & G_0 G_1 G_2 \cdots G_m & & \\ & & G_0 G_1 G_2 \cdots G_m & \\ & & \cdot & \cdot \\ & & \cdot & \cdot \\ & & \cdot & \cdot \end{bmatrix}$$

矩阵中的元素 G_i，$0 \leqslant i \leqslant 8$，可通过 $g_1^{(1)}$ 和 $g_1^{(2)}$ 由如下方法得到：

$G_0 = 11, G_1 = 10, G_2 = 11, G_3 = 11, G_4 = 01, G_5 = 10, G_6 = 00, G_7 = 10, G_8 = 11$

因此，编码符号 c 可按如下方式计算：

$c = d \cdot \boldsymbol{G}$
=（1001 1101 0011 1000 1010 0101）

$$
\begin{bmatrix}
\text{11 10 11 11 01 10 00 10 11} \\
\quad\text{11 10 11 11 01 10 00 10 11} \\
\quad\quad\text{11 10 11 11 01 10 00 10 11} \\
\quad\quad\quad\text{11 10 11 11 01 10 00 10 11} \\
\quad\quad\quad\quad\text{11 10 11 11 01 10 00 10 11} \\
\quad\quad\quad\quad\quad\text{11 10 11 11 01 10 00 10 11} \\
\quad\quad\quad\quad\quad\quad\text{11 10 11 11 01 10 00 10 11} \\
\quad\quad\quad\quad\quad\quad\quad\text{11 10 11 11 01 10 00 11 11} \\
\quad\quad\quad\quad\quad\quad\quad\quad\text{11 10 11 11 01 10 00 11 11} \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\text{11 10 11 11 01 10 00 11 11} \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\text{11 10 11 11 01 10 00 11 11} \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\text{11 10 11 11 01 10 00 11 11} \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\text{11 10 11 11 01 10 00 11 11} \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\text{11 10 11 11 01 10 00 11 11} \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\text{11 10 11 11 01 10 00 11 11} \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\text{11 10 11 11 01 10 00 11 11} \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\text{11 10 11 11 01 10 00 11 11} \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\text{11 10 11 11 01 10 00 11 11} \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\text{11 10 11 11 01 10 00 11 11} \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\text{11 10 11 11 01 10 00 11 11} \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\text{11 10 11 11 01 10 00 11 11} \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\text{11 10 11 11 01 10 00 11 11} \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\text{11 10 11 11 01 10 00 11 11} \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\text{11 10 11 11 01 10 00 11 11}
\end{bmatrix}
$$

$= (11\ 10\ 11\ 00\ 00\ 00\ 10\ 00\ 01\ 00\ 00\ 01\ 01\ 01\ 11\ 11\ 00\ 10\ 01\ 00\ 01\ 10)$

这就是所要的编码符号 c。

3．多项式矩阵方法

信息多项式可表示如下

$$d(D) = D^{23} + D^{20} + D^{19} + D^{18} + D^{16} + D^{13} + D^{12} + D^{11} + D^{7} + D^{5} + D^{2} + 1$$

生成多项式矩阵可表示如下

$$
\begin{aligned}
G(D) &= [g_1^{(1)}(D), g_1^{(2)}(D)] \\
&= [D^8 + D^7 + D^6 + D^5 + D^3 + D + 1, D^8 + D^6 + D^5 + D^4 + 1]
\end{aligned}
$$

在模 2 加法器处的编码符号多项式矩阵由下式获得：

$$
\begin{aligned}
c'(D) &= d(D)G(D) = [c_0(D), c_1(D)] \\
&= [D^{31} + D^{30} + D^{29} + D^{25} + D^{17} + D^{16} + D^{14} + D^{10} + D^9 + D^7 + D^5 + D^2 + D + 1, \\
&\quad D^{31} + D^{29} + D^{23} + D^{20} + D^{19} + D^{18} + D^{17} + D^{16} + D^{13} + D^{11} + D^9 + D^4 + D^2 + 1]
\end{aligned}
$$

因此，由转向开关产生延时而获得的编码输出如下：

$$
\begin{aligned}
c(D) &= Dc_0(D^2) + c_1(D^2) \\
&= D(D^{62} + D^{60} + D^{58} + D^{50} + D^{34} + D^{32} + D^{28} + D^{20} + D^{18} + D^{14} + D^{10} + D^4 + D^2 + 1) \\
&\quad + (D^{62} + D^{58} + D^{46} + D^{40} + D^{38} + D^{36} + D^{34} + D^{32} + D^{26} + D^{22} + D^{18} + D^8 + D^4 + 1) \\
&= D^{63} + D^{62} + D^{61} + D^{59} + D^{58} + D^{51} + D^{46} + D^{40} + D^{38} + D^{36} + D^{35} + D^{34} + D^{33} \\
&\quad + D^{32} + D^{29} + D^{26} + D^{22} + D^{21} + D^{19} + D^{18} + D^{15} + D^{11} + D^8 + D^5 + D^4 + D^3 + D + 1
\end{aligned}
$$

$c = (11\ 10\ 11\ 00\ 00\ 00\ 10\ 00\ 01\ 00\ 00\ 01\ 01\ 01\ 11\ 11$

00 10 01 00 01 10 11 00 10 00 10 01 00 11 10 11）

由此，可以证明以上提及的替代方法都可以获得需要的输出，也方便了读者对这些替代方法的理解。

4.3.2 编码符号重复

同步信道的发送速率固定为 1 200 b/s。对同步信道而言，如表 4.5 所示，每个卷积编码符号在交织前会重复一次（1 个符号连续出现 2 次）。

4.3.3 块交织操作

同步信道中的编码符号在重复一次后进入块交织，目的是为了避免多径衰落环境发送产生突发错误以及为了提高性能而提供接入冗余。

同步信道分割成 80 ms 的超帧。因为同步信道帧由 32 位和 26.667（ = 32/1200）ms 组成，所有超帧分割成 3 个 26.667 的帧。同步信道使用的块交织跨度是 26.667 ms，等同于以符号速率 4 800 s/s 调制 128（ = 16 × 8）个信号。注意到每个编码符号的重复也是调制信号。

同步信道交织器的输入输出在表 4.5 和 4.6 中列出。输入符号表（表 4.5）读法是按列从左到右，第 1 个输入符号（1）是左上角的符号，第 2 个符号（1）是紧跟在第 1 个符号下面的符号，第 17 个输入符号（9）是第 1 个输入符号右边的符号，第 18 个输入符号（9）是在第 2 个输入符号右边的符号。在输入符号表中，同样数字的符号显示了编码符号的重复。输出符号表（表 4.6）读法与输入符号表读法一致，即第 1 个输出符号（1）位于左上角，第 2 个输出符号（33）位于第 1 个输出符号的下面，第 17 个输出符号（3）位于第 1 个输出符号的右边。

表 4.5　同步信道交织器的输入符号表(**TIA/EIA/IS-95**)

1	9	17	25	33	41	49	57
1	9	17	25	33	41	49	57
2	10	18	26	34	42	50	58
2	10	18	26	34	42	50	58
3	11	19	27	35	43	51	59
3	11	19	27	35	43	51	59
4	12	20	28	36	44	52	60
4	12	20	28	36	44	52	60
5	13	21	29	37	45	53	61
5	13	21	29	37	45	53	61
6	14	22	30	38	46	54	62
6	14	22	30	38	46	54	62
7	15	23	31	39	47	55	63
7	15	23	31	39	47	55	63
8	16	24	32	40	48	56	64
8	16	24	32	40	48	56	64

表 4.6　同步信道交织器的输出符号表(**TIA/EIA/IS-95**)

1	3	2	4	1	3	2	4
33	35	34	36	33	35	34	36
17	19	18	20	17	19	18	20
49	51	50	52	49	51	50	52
9	11	10	12	9	11	10	12
41	43	42	44	41	43	42	44
25	27	26	28	25	27	26	28
57	59	58	60	57	59	58	60
5	7	6	8	5	7	6	8
37	39	38	40	37	39	38	40
21	23	22	24	21	23	22	24
53	55	54	56	53	55	54	56
13	15	14	16	13	15	14	16
45	47	46	48	45	47	46	48
29	31	30	32	29	31	30	32
61	63	62	64	61	63	62	64

例 4.6　输入同步信道交织器的序列由卷积编码重复而来。根据表 4.5,所有编码符号重复后会形成如下所示的 128 位符号序列:

1111110011110000000000001100000000001100000000110011001111111111

0000110000110000001111001111000011000000110000110000111111001111

例 4.7　讨论对同步信道交织器输出符号的计算。一般来说,卷积编码符号都会按如下方式排列:

1110110000001000

0100000101011111

0010010001101100

1000100100111011

输出序列是按列从左到右读出。这里只指出表 4.6 中的第 1 列、第 2 列的读法。

交织器输出顺序	相应的符号
1,33,17,49,9,41,25,57, 5,37,21,53,13,45,29,61	1001000010011111
3,35,19,51,11,43,27,59 7,39,23,55,15,47,31,63	1100010100000011

利用表 4.6,可得到交织器第 3 列到第 8 列的输出序列,交织器输出符号序列的完整列表如下:

列号	交织器输出
1	1001000010011111
2	1100010100000011
3	1010011011000110
4	0000001100110011
5	1001000010011111
6	1100010100000011
7	1010011011000110
8	0000001100110011

4.3.4 利用 **Walsh** 函数正交相位扩展

前向信道中要发送的同步信号要由固定码率为 1.228 8 Mc/s 的 Walsh 函数正交相位扩展。

同步信道用 Walsh 函数 32(是 64 个正交 Walsh 函数中的一个)扩展,由此被称为 32 路信道(W_{32})。

在表 4.7 中的首行或首列中,同步信道扩频的 Walsh 函数序号都是由 $n = 32$ 指定的扩展 Walsh 函数。Walsh 函数扩展序列以周期 52.083 μs(= 64/1.2288 Mc/s)重复。

编号为 32 的 Walsh 码片是

00000000 00000000 00000000 00000000 11111111 11111111 11111111 11111111

因为交织器输出 128 位,因此序号为 32 的 Walsh 函数的码片要重复一次。考虑到调制信号符号速率为 4 800 s/s,PN 码率 1.228 8Mc/s,这两个速率之比为 256,刚好等于一个跨度为 256 的交织器的一次输出。

因此,与序号为 32 的 Walsh 函数进行"异或"操作可得到正交调制输出,如下所示:

(a) 11…1 11…1 11…1 11…1 11…1 11…1 11…1 11…1

(b) 00…0 11…1 00…0 11…1 00…0 11…1 00…0 11…1

(c) 11…1 00…0 11…1 00…0 11…1 00…0 11…1 00…0

其中(a):交织器的第一个输出符号 1,与 256 个 Walsh 码片进行"异或"操作,重复 256 次。

(b):Walsh 码片($64 \times 4 = 256$ 位)等于序号为 32 的 Walsh 函数的 4 倍。

(c):对(a),(b)进行"异或"操作,即(a)$\oplus$(b),获得调制输出。

表 4.7　**64 阶 Walsh 函数**

	一个 **Walsh** 指数内的 **Walsh** 码片
	1111111111222222222233333333334444444444555555555566666
	0123456789012345678901234567890123456789012345678901234567890123
0	0000000000000000000000000000000000000000000000000000000000000000
1	0101010101010101010101010101010101010101010101010101010101010101
2	0011001100110011001100110011001100110011001100110011001100110011
3	0110011001100110011001100110011001100110011001100110011001100110
4	0000111100001111000011110000111100001111000011110000111100001111
5	0101101001011010010110100101101001011010010110100101101001011010
6	0011110000111100001111000011110000111100001111000011110000111100
7	0110100101101001011010010110100101101001011010010110100101101001
8	0000000011111111000000001111111100000000111111110000000011111111
9	0101010110101010010101011010101001010101101010100101010110101010
10	0011001111001100001100111100110000110011110011000011001111001100
11	0110011010011001011001101001100101100110100110010110011010011001
12	0000111111110000000011111111000000001111111100000000111111110000
13	0101101010100101010110101010010101011010101001010101101010100101
14	0011110011000011001111001100001100111100110000110011110011000011
15	0110100110010110011010011001011001101001100101100110100110010110
16	0000000000000000111111111111111100000000000000001111111111111111
17	0101010101010101101010101010101001010101010101011010101010101010
18	0011001100110011110011001100110000110011001100111100110011001100
19	0110011001100110100110011001100101100110011001101001100110011001
20	0000111100001111111100001111000000001111000011111111000011110000
21	0101101001011010101001011010010101011010010110101010010110100101
22	0011110000111100110000111100001100111100001111001100001111000011
23	0110100101101001100101101001011001101001011010011001011010010110
24	0000000011111111111111110000000000000000111111111111111100000000
25	0101010110101010101010100101010101010101101010101010101001010101
26	0011001111001100110011000011001100110011110011001100110000110011
27	0110011010011001100110010110011001100110100110011001100101100110
28	0000111111110000111100000000111100001111111100001111000000001111
29	0101101010100101101001010101101001011010101001011010010101011010
30	0011110011000011110000110011110000111100110000111100001100111100
31	0110100110010110100101100110100101101001100101101001011001101001
32	0000000000000000000000000000000011111111111111111111111111111111
33	0101010101010101010101010101010110101010101010101010101010101010
34	0011001100110011001100110011001111001100110011001100110011001100
35	0110011001100110011001100110011010011001100110011001100110011001
36	0000111100001111000011110000111111110000111100001111000011110000
37	0101101001011010010110100101101010100101101001011010010110100101
38	0011110000111100001111000011110011000011110000111100001111000011
39	0110100101101001011010010110100110010110100101101001011010010110
40	0000000011111111000000001111111111111111000000001111111100000000
41	0101010110101010010101011010101010101010010101011010101001010101
42	0011001111001100001100111100110011001100001100111100110000110011
43	0110011010011001011001101001100110011001011001101001100101100110
44	0000111111110000000011111111000011110000000011111111000000001111
45	0101101010100101010110101010010110100101010110101010010101011010
46	0011110011000011001111001100001111000011001111001100001100111100
47	0110100110010110011010011001011010010110011010011001011001101001
48	0000000000000000111111111111111111111111111111110000000000000000

（续表）

	一个 **Walsh** 指数内的 **Walsh** 码片
	1111111111222222222233333333334444444444555555555566666
	0123456789012345678901234567890123456789012345678901234567890123
49	0101010101010101101010101010101010101010101010010101010101010101
50	0011001100110011110011001100110011001100110011000011001100110011
51	0110011001100110100110011001100110011001100110011001100110011001
52	0000111100001111111100001111000011110000111100000000111100001111
53	0101101001011010101001011010010110100101101001010101101001011010
54	0011110000111100110000111100001111000011110000110011110000111100
55	0110100101101001100101101001011010010110100101100110100101101001
56	0000000011111111111111110000000011111111000000000000000011111111
57	0101010110101010101010100101010110101010010101010101010110101010
58	0011001111001100110011000011001111001100001100110011001111001100
59	0110011010011001100110010110011010011001011001100110011010011001
60	0000111111110000111100000000111111110000000011110000111111110000
61	0101101010100101101001010101101010100101010110100101101010100101
62	0011110011000011110000110011110011000011001111000011110011000011
63	0110100110010110100101100110100110010110011010010110100110010110

因此，正交调制信号计算如下：

11111111 11111111 11111111 11111111 00000000 00000000 00000000 00000000

11111111 11111111 11111111 11111111 00000000 00000000 00000000 00000000

11111111 11111111 11111111 11111111 00000000 00000000 00000000 00000000

11111111 11111111 11111111 11111111 00000000 00000000 00000000 00000000

正交相位扩展后，同步信道将会进行如图 4.10 所示的正交相位扩展。

4.3.5 正交相位扩展

在 4.2.2 一节中讨论了用 I 路和 Q 路导频 PN 序列来扩频同步信道。在下面例子中，显示如何利用 LFSR 二进制序列 $i(n)$ 和 $q(n)$ 产生导频 PN 序列 $\overline{P}_I$ 和 $\overline{P}_Q$。

例 4.8 使用 $i(n)$，$q(n)$ 和 LFSR 的初始内容，即 IC =（1000000000000000），同相导频 PN 序列 P_I 和正交相位导频 PN 序列 $\overline{P}_Q$ 计算如下：

（Ⅰ）利用 $i(n)=i(n-15)=0$，$1\leqslant n\leqslant 14$，$i(15)=1$，$i(n)=\sum_{k=15,10,8,7,6,2} i(n-k)$，$16\leqslant n\leqslant 32\,767$，在 $i(n)$ 中连续出现 14 个 0 后的位置（参见箭头↓）插入 0，就可以得到同相导频 PN 序列 $\overline{P}_I$，如下图所示。

插入 0

↓

P_I = 00000000000000001010100100111010001101111001100100
0

0011110000100011010010101101010101011101001001111 1

注意计算是在 $0\leqslant n\leqslant 99$ 范围内进行的。

（Ⅱ）类似地，利用 $q(n)=q(n-15)=0$，$1\leqslant n\leqslant 14$，$q(15)=1$，$q(n)=\sum_{k=15,12,11,10,9,5,4,3} q(n-k)$，$16\leqslant n\leqslant$ 32 767，在 $q(n)$ 中连续出现第 15 个 0 的位置插入 0 就可以得到正交相位导频 PN 序列 $\overline{P}_Q$，如下图

所示。

插入 0



$\bar{P}_Q =$ 000000000000000100111101011101011010011100010100111
100111000110110001110100110001001001000110011000111

例 4.9　同相序列 I 和正交相位序列 Q 由如下扩频操作获得：

$I = \mathrm{OSC} \oplus \bar{P}_I$ 和 $Q = \mathrm{OSC} \oplus \bar{P}_Q$

其中 OSC 表示正交符号覆盖。

OSC：　11111111 11111111 11111111 11111111 00000000…

$\oplus$

P_I　　00000000 00000001 01010010 01110100 01101111…

I：　　11111111 11111110 10101101 10001011 01101111…

和

OSC：　11111111 11111111 11111111 11111111 00000000…

$\oplus$

$\bar{P}_Q$　　00000000 00000001 00111101 01110101 10100111…

Q：　　11111111 11111110 11000010 10001010 10100111…

经过扩频后，I 和 Q 脉冲将用做如图 4.10 所示的 I 路和 Q 路基带滤波的输入。

4.3.6　同步信道滤波

在 4.2.3 一节中提到了同步信道滤波。基带滤波器的频率响应 $S(f)$ 满足图 13.15 的限定。

特别地，滤波器的归一化频率响应通带 $0 \leqslant f \leqslant f_p$ 限定在 $\pm\delta_1$，阻带 $f \geqslant f_s$ 小于或等于 $-\delta_2$，参数值为 $\delta_1 = 1.5$ dB，$\delta_2 = 40$ dB，$f_p = 590$ KHz 和 $f_s = 740$ kHz。

4.3.7　正交相移键控

基带滤波后，同相序列 $I(t)$ 和正交相序列 $Q(t)$ 分别调制到余弦和正弦载波上形成两路 BPSK 波形。两路 BPSK 波形步调一致，它们的相位在每个 T_b 时刻变化。

如图 4.10 所示，QPSK 波形 $s(t)$ 是正交的 BPSK 波形 $I(t)$ 与 $Q(t)$ 之和，前向编码信道的标准 QPSK 已在 4.2.4 中讨论。

例 4.10　由以上计算可知，同相序列 I 和正交相序列 Q 分别是

I：	11111111	11111110	10101101	10001011	01101111
Q：	11111111	11111110	11000010	10001010	10100111

考虑两序列的一部分，$I(t)$ = 10101101，$Q(t)$ = 11000010。由 $I(t)$ 和 $Q(t)$ 组合的 QPSK 波形 $s(t)$ 如图 4.13 所示。

4.3.8　同步信道信令和消息结构

这部分主要讨论同步信道上发送消息的要求和格式。

同步信道分割成 80 ms，96 位的超帧，超帧又分割成 3 个 26.667（ = 32/1200 b/s）ms 的帧（因

为同步信道在固定速率为 1 200 b/s 下工作)，每一帧的第一位是消息开始位(SOM 位)，帧中其他的位组成同步信道的帧体。

同步信道(1 200 b/s)

I 信道输出：10101101(－1，1，－1，1，－1，－1，1，－1)

Q 信道输出：11000010(－1，－1，1，1，1，1，－1，1)

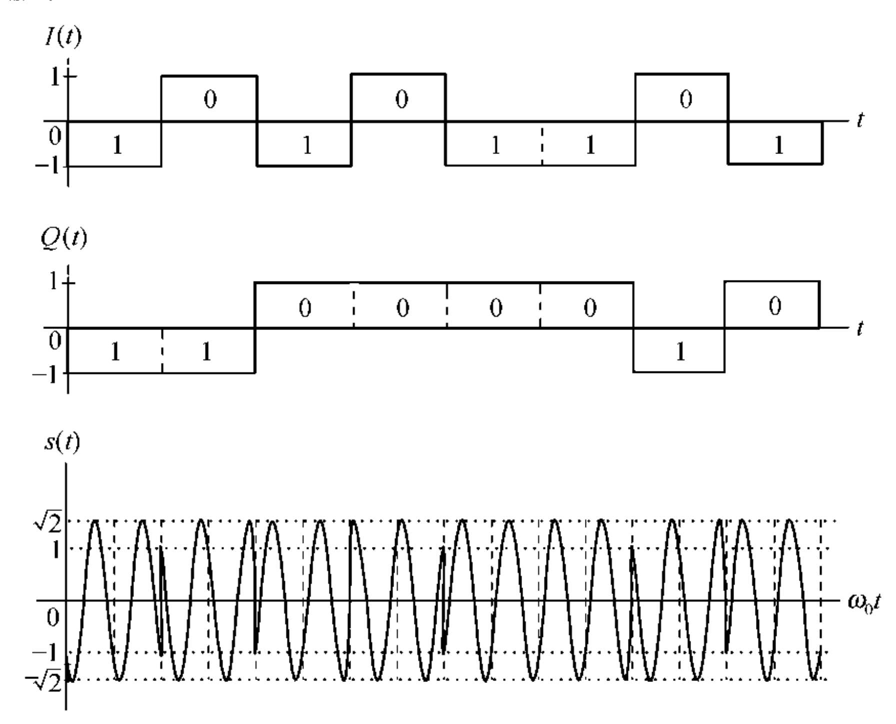
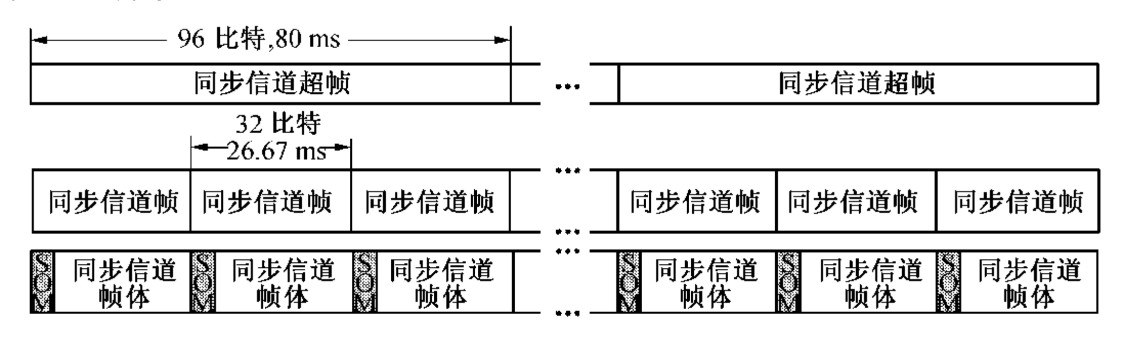
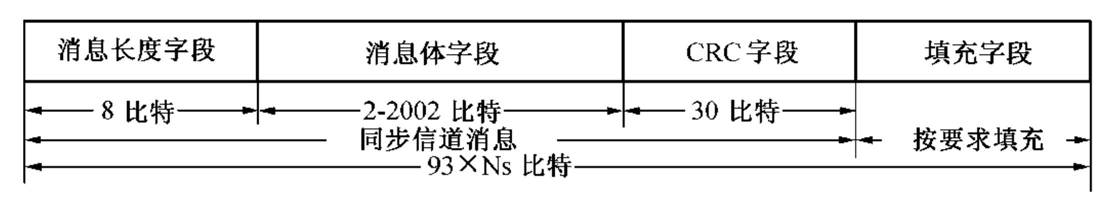


图 4.13　同步信道 QPSK 波形 $s(t)$

同步信道信息是由同步信道消息和填充组成，包括一个 8 比特的消息长度字段、消息体字段、30 比特的循环冗余校验(CRC)字段。在消息末尾的填充位设为 0。一个典型的同步信道结构(1 200 b/s)如图 4.14 所示。



(a) 同步信道消息封装



(b) 同步信道消息＋填充

SOM:消息比特的开始　Ns:传送消息所需的同步信道超帧数目

图 4.14　1 200 b/s 同步信道结构的典型例子

同步信道消息封装从同步信道超帧的第一个同步帧体的首位开始。基站将同步信道消息封装的第一个消息的 SOM 位设为 1,其余的设为 0,如图 4.14 所示。

基站将同步信道消息放到连续信道帧体中发送。基站在同步信道消息封装中加入足够的填充位,直到下一个超帧的 SOM 位出现,基站将在超帧的第一个帧中开始新的同步信道消息封装。基站限定的最大同步消息长度是 148(8 进制),即 148 × 8 = 1184 比特。

4.3.9　同步信道的循环冗余校验(CRC)

在同步信道发送消息中要计算 30 比特的 CRC。

CRC 生成多项式如下:

$$g(x)=1+x+x^2+x^6+x^7+x^8+x^{11}+x^{12}+x^{13}+x^{15}+x^{20}+x^{21}+x^{29}+x^{30} \tag{4.13}$$

其逻辑图如图 4.15。

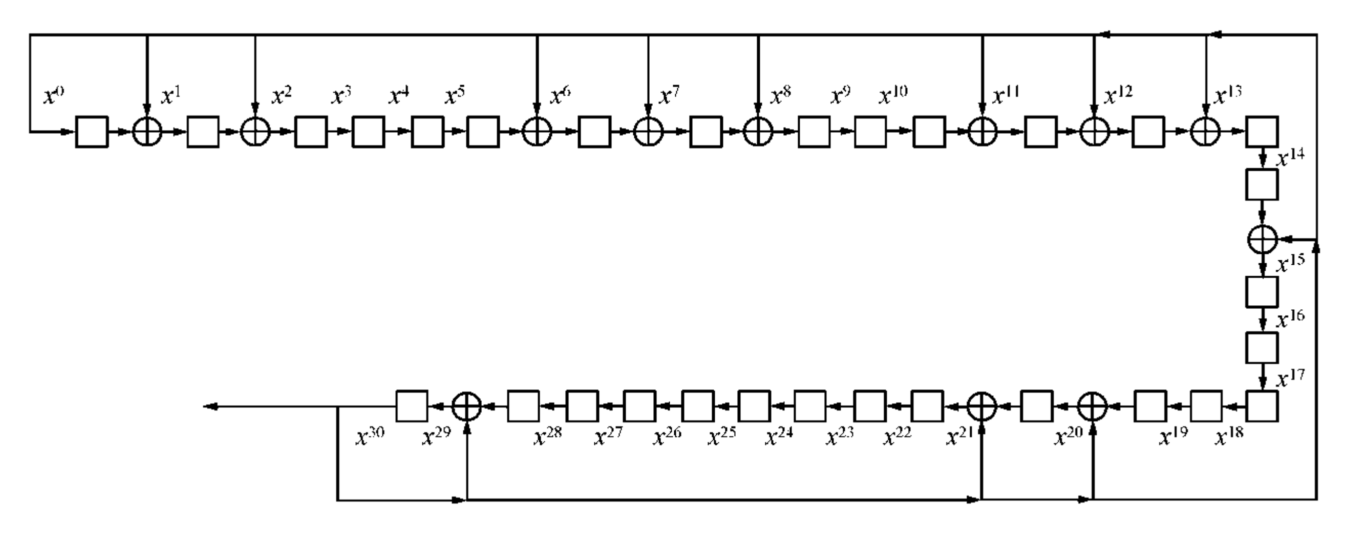


图 4.15 LFSR 逻辑图的生成多项式表示

$$g(x)=1+x+x^2+x^6+x^7+x^8+x^{11}+x^{12}+x^{13}+x^{15}+x^{21}+x^{29}+x^{30}$$

下面讨论计算 CRC 的具体过程,其编码逻辑图如图 4.16 所示。

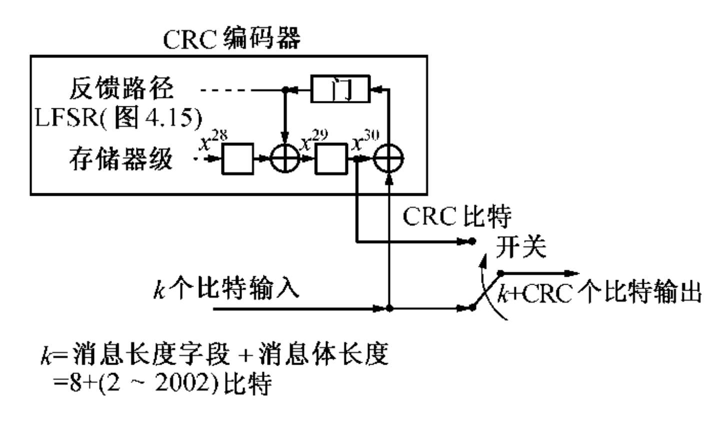


图 4.16　同步信道 CRC 编码器

1. LFSR 的初始内容置为 1,目的是即使全 0 输入 CRC 也不会为 0。
2. 将开关置下,合上 CRC 编码器的反馈路径。
3. k 比特输入数据发送给编码器输出,以及移位寄存器的反馈路径。

k 比特输入包括 8 比特消息长度字段和消息体。

4. 输入 k 位过程中寄存器关闭 k 次。
5. 将开关置上，打开 CRC 编码器的反馈路径。
6. 再关闭寄存器 30 次，CRC 字段将产生。
7. 因此，同步信道消息以 8 位消息长度字段、消息体、30 位 CRC 字段的顺序从编码器中输出。

4.4 寻呼信道

寻呼信道是经过编码、交织、扩频以及调制扩频的信号，供基站覆盖范围内的移动台使用。基站利用寻呼信道发送系统开销信息和移动台专用消息。信号信道发送的信息速率固定为9 600 b/s或4 800 b/s，帧长为 20 ms。速率为 9.6 kb/s 或 4.8 kb/s 的寻呼信道结构如图 4.18 所示。

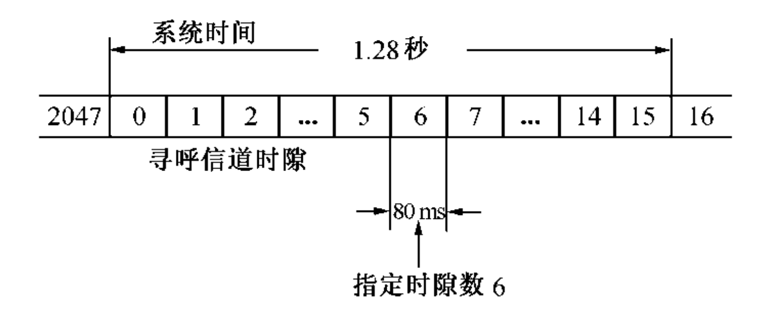


图 4.17 移动台时隙模式操作

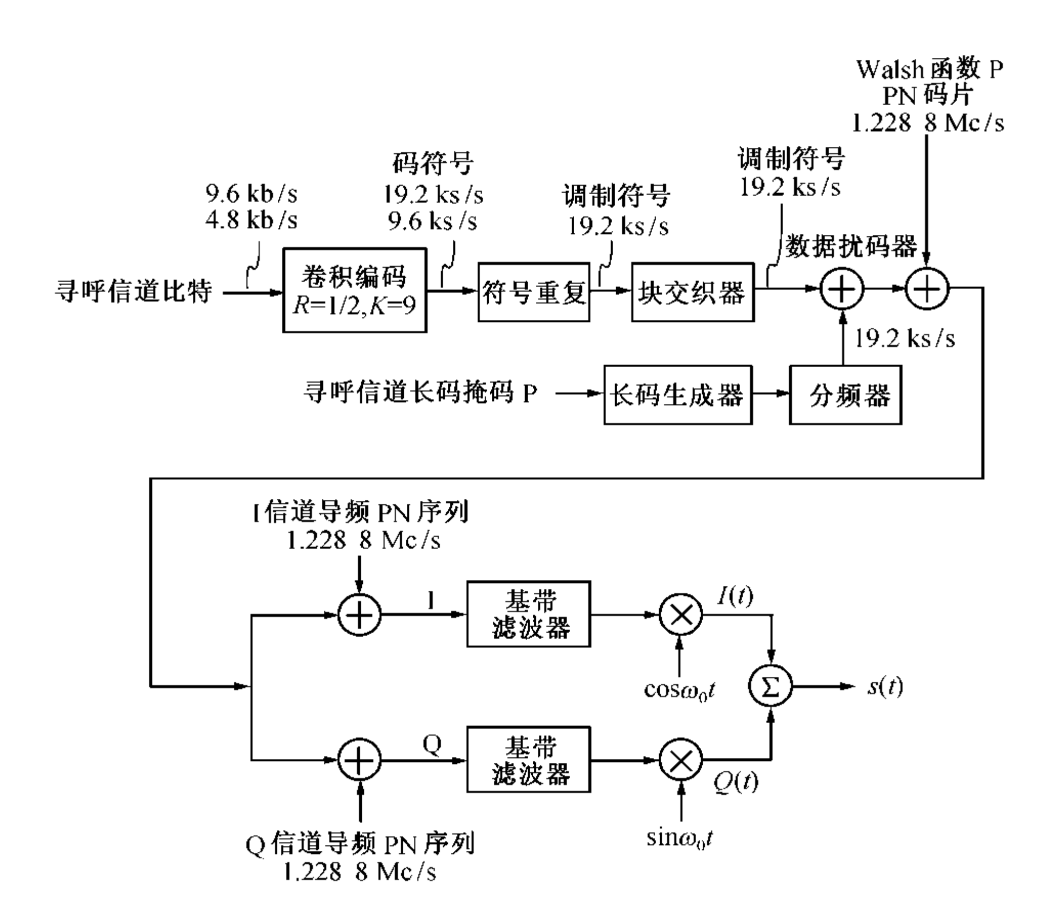


图 4.18 数据速率为 9 600 b/s 或 4 800 b/s 的寻呼信道结构

寻呼信道的导频 PN 序列也是由同样的 PN 序列移位得到，作为基站的导频信道。

第一个寻呼信道帧首先被基站发送出去，寻呼信道每80 ms被分成寻呼信道时隙。

采用非时隙模式操作，移动台在所有寻呼信道时隙上都会收到寻呼和控制消息，采用时隙模式操作，移动台只在指定寻呼时隙上检测信道。如图4.17所示，时隙长为1.28秒，计算得到的寻呼时隙数是6，因此，移动台时隙周期地始于时隙6。移动台将在时隙6开始时开始检测寻呼信道。最短的时隙周期由16个80 ms的时隙组成，图4.17中，时长为1.28（ = 16×80 ms）秒。

寻呼信道时隙数计算如下：

$$时隙数 = \lfloor t/4 \rfloor \bmod 2048$$

其中t是帧中系统时间，mod 2048代表最大长度时隙周期（2 048时隙）。寻呼信道调制参数见表4.8。

表4.8　寻呼信道调制参数

参　数	数据速率（b/s）		
	9 600	4 800	单位
PN码片速率	1.228 8	1.228 8	Mc/s
码率	1/2	1/2	比特/编码符号
码重复	1	2	调制符号/编码符号
调制符号速率	19 200	19 200	s/s
PN码片/调制符号	64	64	PN码片/调制符号
PN码片/比特	128	256	PN码片/比特

4.4.1　寻呼信道编码

在4.3.1一节中讨论了在发送之前，寻呼信道数据进行卷积编码，（2.1.8）的卷积码应用于寻呼、同步、前向业务信道中，如图4.12所示的编码结构也适用于寻呼信道。寻呼信道的卷积编码生成序列是$g_1^{(1)} = (111101011)$和$g_1^{(2)} = (101110001)$。因为卷积编码器对时延数据序列选择的抽头进行模2加，生成多项式可写做如下形式：

$$g_1^{(1)}(x) = 1 + x + x^2 + x^3 + x^5 + x^7 + x^8$$

和

$$g_1^{(2)}(x) = 1 + x^2 + x^3 + x^4 + x^8$$

因为码率为1/2，对应一个输入的两个编码符号输出分别是由$g_1^{(1)}$编码得到的符号c_0以及由$g_1^{(2)} = (101110001)$编码得到的符号c_1。

卷积编码器初始化后输出的第一个符号由生成序列$g_1^{(1)}$生成。

例4.11　考虑（2,1,8）卷积编码器，其生成多项式$g_1^{(1)} = (111101011)$和$g_1^{(2)} = (101110001)$如图4.12所示。选择寻呼信道操作固定速率为4 800 b/s。

给定96比特的信息序列

d = (1100111010 0011011000 0011100011 0100101000 1010010111
0101010011 1100101001 0110011101 0111000010 110111)

编码器初始化内容为全零，编码符号c_0和c_1分别由下式生成：

$$c_0 = d_0 + d_1 + d_2 + d_3 + d_5 + d_7 + d_8$$

$$c_1 = d_0 + d_2 + d_3 + d_4 + d_8$$

由$g_1^{(1)}$和$g_1^{(2)}$生成的96位符号序列c_0和c_1分别是

c_0 = (1000010010 0111100110 1000010000 0110101101 1111011000

1100001001 1100111010 1001000101 0111011011 000011）

c_1 =（1110100011 0110010000 0110101001 0010011000 0011101011

1000100101 0010111011 1010110101 1100100011 110111）

一对串联复接（c_0，c_1）产生下列 192 位编码器输出符号序列 c：

c =（1101010001100000

1101001111101001

0010100010010100

0110010000010010

1100100111100010

1010111101101100

0101111000000100

1001001110100100

1111110011011100

0110010100110011

0111101001101000

1111010100011111）

c 的排列形式是 12 行 16 列，目的是便于利用表 4.10 查找寻呼信道交织器输出。

4.4.2 符号重复和块交织

如果信息速率低于 9 600 b/s，则经过卷积编码器的符号在进行块交织之前要重复。如果速率是 4 800 b/s，每个符号重复一次，即每个符号连续发送两次。这样，可保证调制符号速率为 19.2 ks/s。

经过重复后，所有符号要进行交织。寻呼和前向业务信道使用的交织器跨度为 20 ms，等于调制符号速率为 19.2 ks/s 时的 384 个调制符号。数据速率为 4 800 b/s 和 9 600 b/s 的输入、输出符号如表 4.9、表 4.10、表 4.11 和表 4.12 所示，表的读法与同步信道交织器一样按列从左到右读。

利用表 4.12，384 位交织器输出如下：

1010101011010110 1010010001111100 1000000011101001 0100110110111010

0110111000010110 1100001000010011 1011111100111010 1110100010011001

1011010110001101 1001111001010100 1001001100100010 1111000000100011

1010101011010110 1010010001111100 1000000011101001 0100110110111010

0110111000010110 1100001000010011 1011111100111010 1110100010011001

1011010110001101 1001111001010100 1001001100100010 1111000000100011

在符号序列中，相同值的符号为码符号的重复。

对 9 600 b/s 的寻呼信道，卷积编码器结果在块交织前不用重复，速率为 9 600 b/s 交织的输入和输出如表 4.11 和表 4.12 所示。速率为 9 600 b/s 的寻呼信道使用的交织器的跨度也是 20 ms，等于调制符号速率为 19.2 ks/s 的 384 个调制符号。

在 CDMA 信道中，符号重复和块交织的先后顺序并不重要。换句话说，卷积编码符号可能在符号重复前进行块交织。

表 4.9　速率为**4 800 b/s** 的寻呼信道和前向业务信道交织输入(**TIA/EIA/IS-95**)

1	13	25	37	49	61	73	85	97	109	121	133	145	157	169	181
1	13	25	37	49	61	73	85	97	109	121	133	145	157	169	181
2	14	26	38	50	62	74	86	98	110	122	134	146	158	170	182
2	14	26	38	50	62	74	86	98	110	122	134	146	158	170	182
3	15	27	39	51	63	75	87	99	111	123	135	147	159	171	183
3	15	27	39	51	63	75	87	99	111	123	135	147	159	171	183
4	16	28	40	52	64	76	88	100	112	124	136	148	160	172	184
4	16	28	40	52	64	76	88	100	112	124	136	148	160	172	184
5	17	29	41	53	65	77	89	101	113	125	137	149	161	173	185
5	17	29	41	53	65	77	89	101	113	125	137	149	161	173	185
6	18	30	42	54	66	78	90	102	114	126	138	150	162	174	186
6	18	30	42	54	66	78	90	102	114	126	138	150	162	174	186
7	19	31	43	55	67	79	91	103	115	127	139	151	163	175	187
7	19	31	43	55	67	79	91	103	115	127	139	151	163	175	187
8	20	32	44	56	68	80	92	104	116	128	140	152	164	176	188
8	20	32	44	56	68	80	92	104	116	128	140	152	164	176	188
9	21	33	45	57	69	80	93	105	117	129	141	153	165	177	189
9	21	33	45	57	69	81	93	105	117	129	141	153	165	177	189
10	22	34	46	58	70	82	94	106	118	130	142	154	166	178	190
10	22	34	46	58	70	82	94	106	118	130	142	154	166	178	190
11	23	35	47	59	71	83	95	107	119	131	143	155	167	179	191
11	23	35	47	59	71	83	95	107	119	131	143	155	167	179	191
12	24	36	48	60	72	84	96	108	120	132	144	156	168	180	192
12	24	36	48	60	72	84	96	108	120	132	144	156	168	180	192

表 4.10　速率为**4 800 b/s** 的寻呼和前向业务信道交织器输出(**TIA/EIA/IS-95**)

1	5	3	7	2	6	4	8	1	5	3	7	2	6	4	8
33	37	35	39	34	38	36	40	33	37	35	39	34	38	36	40
65	69	67	71	66	70	68	72	65	69	67	71	66	70	68	72
97	101	99	103	98	102	100	104	97	101	99	103	98	102	100	104
129	133	131	135	130	134	132	136	129	133	131	135	130	134	132	136
161	165	163	167	162	166	164	168	161	165	163	167	162	166	164	168
17	21	19	23	18	22	20	24	17	21	19	23	18	22	20	24
49	53	51	55	50	54	52	56	49	53	51	55	50	54	52	56
81	85	83	87	82	86	84	88	81	85	83	87	82	86	84	88
113	117	115	119	114	118	116	120	113	117	115	119	114	118	116	120
145	149	147	151	146	150	148	152	145	149	147	151	146	150	148	152
177	181	179	183	178	182	180	184	177	181	179	183	178	182	180	184
9	13	11	15	10	14	12	16	9	13	11	15	10	14	12	16
41	45	43	47	42	46	44	48	41	45	43	47	42	46	44	48
73	77	75	79	74	78	76	80	73	77	75	79	74	78	76	80
105	109	107	111	106	110	108	112	105	109	107	111	106	110	108	112
137	141	139	143	138	142	140	144	137	141	139	143	138	142	140	144
169	173	171	175	170	174	172	176	169	173	171	175	170	174	172	176
25	29	27	31	26	30	28	32	25	29	27	31	26	30	28	32
57	61	59	63	58	62	60	64	57	61	59	63	58	62	60	64
89	93	91	95	90	94	92	96	89	93	91	95	90	94	92	96
121	125	123	127	122	126	124	128	121	125	123	127	122	126	124	128
153	157	155	159	154	158	156	160	153	157	155	159	154	158	156	160
185	189	187	191	186	190	188	192	185	189	187	191	186	190	188	192

表 4.11 速率为9 600 b/s 的寻呼信道和前向业务信道交织器输入(**TIA/EIA/IS-95**)

1	25	49	73	97	121	145	169	193	217	241	265	289	313	337	361
2	26	50	74	98	122	146	170	194	218	242	266	290	314	338	362
3	27	51	75	99	123	147	171	195	219	243	267	291	315	339	363
4	28	52	76	100	124	148	172	196	220	244	268	292	316	340	364
5	29	53	77	101	125	149	173	197	221	245	269	293	317	341	365
6	30	54	78	102	126	150	174	198	222	246	270	294	318	342	366
7	31	55	79	103	127	151	175	199	223	247	271	295	319	343	367
8	32	56	80	104	128	152	176	200	224	248	272	296	320	344	368
9	33	57	81	105	129	153	177	201	225	249	273	297	321	345	369
10	34	58	82	106	130	154	178	202	226	250	274	298	322	346	370
11	35	59	83	107	131	155	179	203	227	251	275	299	323	347	371
12	36	60	84	108	132	156	180	204	228	252	276	300	324	348	372
13	37	61	85	109	133	157	181	205	229	253	277	301	325	349	373
14	38	62	86	110	134	158	182	206	230	254	278	302	326	350	374
15	39	63	87	111	135	159	183	207	231	255	279	303	327	351	375
16	40	64	88	112	136	160	184	208	232	256	280	304	328	352	376
17	41	65	89	113	137	161	185	209	233	257	281	305	329	355	379
18	42	66	90	114	138	162	186	210	234	258	282	306	330	354	378
19	43	67	91	115	139	163	187	211	235	259	283	307	331	355	379
20	44	68	92	116	140	164	188	212	236	260	284	308	332	356	380
21	45	69	93	117	141	165	189	213	237	261	285	309	333	357	381
22	46	70	94	118	142	166	190	214	238	262	286	310	334	358	382
23	47	71	95	119	143	167	191	215	239	263	287	311	335	359	383
24	48	72	96	120	144	168	192	216	240	264	288	312	336	360	384

表 4.12 速率为9 600 b/s 的寻呼信道和前向业务信道交织器输出(**TIA/EIA/IS-95**)

1	9	5	13	3	11	7	15	2	10	6	14	4	12	8	16
65	73	69	77	67	75	71	79	66	74	70	78	68	76	72	80
129	137	133	141	131	139	135	143	130	138	134	142	132	140	136	144
193	201	197	205	195	203	199	207	194	202	198	206	196	204	200	208
257	265	261	269	259	267	263	271	258	266	262	270	260	268	264	272
321	329	325	333	323	331	327	335	322	330	326	334	324	332	328	336
33	41	37	45	35	43	39	47	34	42	38	46	36	44	40	48
97	105	101	109	99	107	103	111	98	106	102	110	100	108	104	112
161	169	165	173	163	171	167	175	162	170	166	174	164	172	168	176
225	233	229	237	227	235	231	239	226	234	230	238	228	236	232	240
289	297	293	301	291	299	295	303	290	298	294	302	292	300	296	304
353	361	357	365	355	363	359	367	354	362	358	366	356	364	360	368
17	25	21	29	19	27	23	31	18	26	22	30	20	28	24	32
81	89	85	93	83	91	87	95	82	90	86	94	84	92	88	96
145	153	149	157	147	155	151	159	146	154	150	158	148	156	152	160
209	217	213	221	211	219	215	223	210	218	214	222	212	220	216	224
273	281	277	285	275	283	279	287	274	282	278	286	276	284	280	288
337	345	341	349	339	347	343	351	338	346	342	350	340	348	344	352
49	57	53	61	51	59	55	63	50	58	54	62	52	60	56	64
113	121	117	125	115	123	119	127	114	122	118	126	116	124	120	128
177	185	181	189	179	187	183	191	178	186	182	190	180	188	184	192
241	249	245	253	243	251	247	255	242	250	246	254	244	252	248	256
305	313	309	317	307	315	311	319	306	314	310	318	308	316	312	320
369	377	373	381	371	379	375	383	370	378	374	382	372	380	376	384

4.4.3　寻呼信道数据扰码

扰码应用于寻呼信道和前向业务信道中，针对块交织器输出的、调制符号速率为19.2 ks/s的信号。

如图 4.18 所示，扰码是将交织器输出结果与长码 PN 码片的 2 进制数模 2 加，长码周期为 $2^{42}-1$，由 LFSR 抽头多项式 $p(x)$生成：

$$p(x)=1+x+x^{2}+x^{3}+x^{5}+x^{6}+x^{7}+x^{10}+x^{16}+x^{17}+x^{19}+x^{21}+x^{22}+x^{25}+x^{26}+x^{27}+x^{31}+x^{33}+x^{35}+x^{22} \quad (4.14)$$

长码中的 PN 码片由 42 位掩码模 2 加生成，抽头序列生成器的 42 位状态向量如图 4.19 所示。

如图 4.20 所示，寻呼信道利用寻呼信道长码的掩码进行扰码。42 位序列组成的长码的掩码是长码的惟一标识，长码将提供有限的私密。

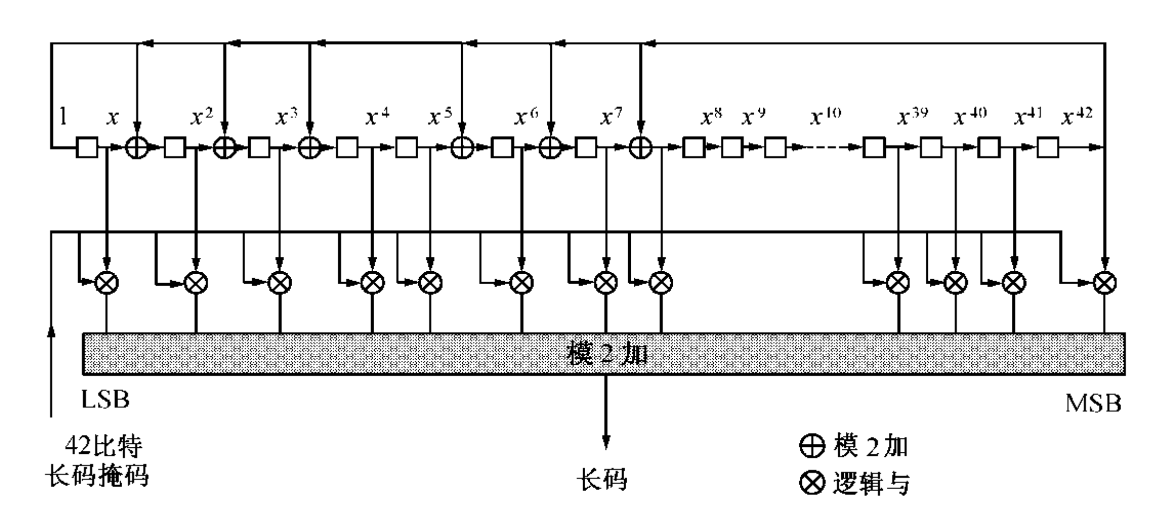
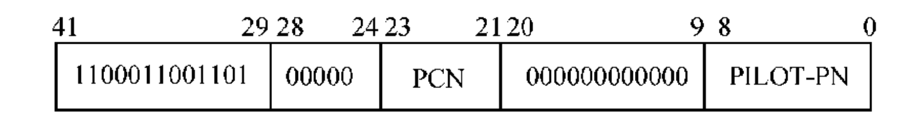


图 4.19　长码生成器



41 … 29	28 … 24	23 … 21	20 … 9	8 … 0
1100011001101	00000	PCN	000000000000	PILOT-PN

PCN：寻呼信道数
PILOT-PN：前向 CDMA 信道的导频序列偏移序号

图 4.20　寻呼信道长码掩码

扰码机制如图 4.21 所示。长码工作时钟为 1.228 8MHz，等于长码生成器的输出 PN 码片序列。长码分割成 64 位的序列，每 64 位中的第一位用于速率为 19.2 ks/s 的扰码。如图 4.21 所示的抽取函数通过每 64($=1.2288\times10^{6}/192\times10^{2}$)位取 1 位减少长码大小。

例 4.12　讨论如图4.21所示的扰码例子。通过对交织器输出和长码的抽取进行"异或"得到扰码数据。利用寻呼信道长码掩码，长码可如下计算。设导频信道数（PCN）是 001、导频 PN 序列偏移序号是 100100011。因此，图 4.20 中的长码掩码序列变为：

110001100110100000001000000000000100100011

利用公式 4.14，长码生成抽头序列可写成：

111101110010000011110110011100010101000000

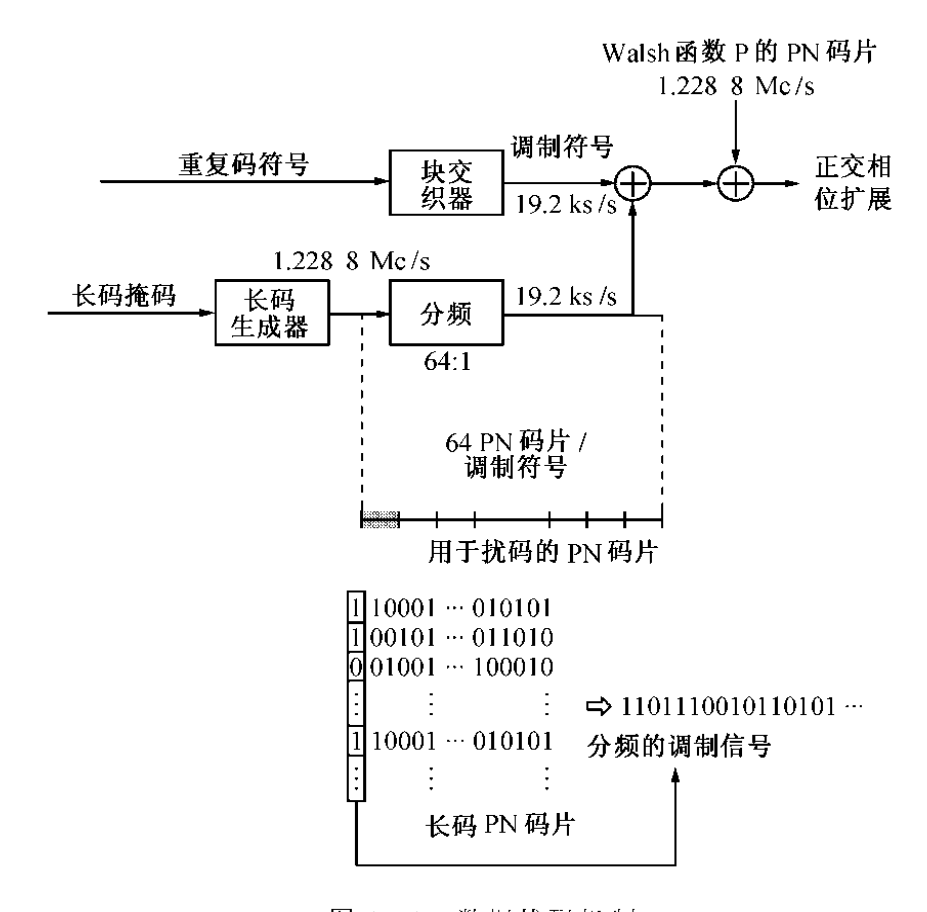


图 4.21　数据扰码机制

初始内容为 10000000000000000000000000000000000000000 时，利用长码掩码和图 4.19 所示的生成抽头序列，可得到长码生成器的输出。

生成的长码序列如下：

1100010010000000 0000010000000101 0001100011011110 1111110100010101
1001011101110001 1100001011100101 0000011111011010 0010101101011010
0010011111111110 0000100110100101 0010010110111000 0001001000100010
1011101001001100 1000111010011101 1001001110010000 0100111001010010
1111001000110010 1111101000010000 0011110111110011 0001100010111011
1010010101111010 0100000111110001 1101010110001110 1110100010010100
0111100110100110 1011000010010100 0011011111000001 1000000011110001
0000001101010000 1101111010111101 0001010101101111 1011110000010100
1101100101101001 0000010010001010 0101100110000101 1001100010111111
0001011100100101 0100010100100100 0001010001100100 0011000101110011
1001111001001000 1001011110010001 0111000000000011 0000101101110010
1101101111101110 0101100011101111 0001000101010001 0101010101111001
0010001110100010 1100100000101011 1010101010011101 1111110100001101
1110000101010011 1010100000001010 1010110010001011 1111010000011100
0101011001100011 0111000111010101 1010011101100110 1000000100011110
1100101001101010 1010101111100110 1110011000000010 1111010000000010

因为每 64 位的第一个输出用于速率为 19.2 ks/s 的数据扰码，与交织器输出符号进行异或操

作的调制符号是 1101110010110101…(最左边一列)。

交织器输出:	1010101011010110…
分频码:	1101110010110101…
EXOR 输出(⊕):	0111011001100011…

符号序列(a)会由 Walsh 函数以固定码片速率 1.228 8 Mc/s 进行扩频,由此让前向信道的所有信道正交。

4.4.4　寻呼信道正交相位扩展

寻呼信道的正交相位扩展使用如表 4.7 定义的 64 个 Walsh 函数中的一个。实际应用中,对寻呼信道 n 使用的扩展序列是 Walsh 函数($1 \leqslant n \leqslant 7$)。调制信号使用由 Walsh 函数产生的 7 个相互正交波形。Walsh 函数扩展序列周期是 52.083 μs(= 64/1.2288 Mc/s),等于寻呼信道调制符号持续时间。

如图 4.1 所示,寻呼信道的编号是从 1 到 7,即 W_1, W_2,…,W_7。

例 4.13　选定寻呼信道1(W_1),相应指数为 1 的 Walsh 函数的 64 位 Walsh 码片如下:

(b)01010101010101010101010101010101010101010101010101
　01010101010101(速率为 1.228 8 Mc/s)

因为 PN 码片/调制符号是 1.2288 Mc/s/19.2 ks/s = 64,与 64 位 Walsh 码片代码相应的调制符号是:

首位:(a):0000000000…00000000
　　(b):0101010101…01010101
异或(a⊕b):0101010101…01010101
第二位(a):1111111111…11111111
　　(b):0101010101…01010101
异或(a⊕b):1010101010…10101010

这里只显示在 19.2 ks/s 速率下对符号序列(a)中前两个符号进行异或操作。类似地,对其他输入符号进行模 2 加会产生如下的 Walsh 符号输出。

01010101 01010101 01010101 01010101 01010101 01010101 01010101 01010101
10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101010
10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101010
10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101010
01010101 01010101 01010101 01010101 01010101 01010101 01010101 01010101
10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101010

4.4.5　寻呼信道正交扩频

正交相位扩展后,寻呼信道进行正交扩频,如图 4.18 所示。15 级的 PN 码片生成器用于正交扩频。导频 PN 序列会在 14 个 0 后添一个 0,导致周期为 2^{15} = 32 768 个码片。

在 4.2.2 一节中，描述了用 I 路和 Q 路序列对寻呼信道进行正交扩频。

例 4.14 同相导频 PN 序列 $\bar{P}_I$ 和正交相序列 $\bar{P}_Q$ 可利用长为 $2^{15}-1$ 的 LFSR 序列 $i(n)$和$q(n)$计算得到。

（Ⅰ）利用 $i(n)=i(n-15)=0\ (1\leqslant n\leqslant 14)$，$i(15)=1$ 和 $i(n)=\sum_{k=15,10,8,7,6,2} i(n-k)$，$16\leqslant n\leqslant 32\ 767$，导频 PN 系列 $\bar{P}_I$ 的计算过程是：先计算得到 $i(n)$，然后在 $i(n)$中连续 14 个零之后插入零，如下所示：

↓

$\bar{P}_I=$ 00000000000000010101001001110100011011110011001000001111000001000110100101011010101010111010010011111

0≤n≤99

（Ⅱ）类似地，利用 $q(n)=q(n-15)=0(1\leqslant n\leqslant 14)$，$q(15)=1$，$q(n)=\sum_{k=15,12,11,10,9,5,4,3} q(n-k)$，$16\leqslant n\leqslant 32\ 767$，正交相导频 PN 序列 $\bar{P}_Q$ 的计算过程是：先计算得到 $q(n)$，然后在 $q(n)$中连续 14 个 0 之后插入 0，如下所示：

↓

$P_Q=$ 00000000000000010011110101110101101001110001010011
10011100011011000111010011000100100100011001100011

0≤n≤99

同相信道序列 I 和正交相信道序列 Q 由如下操作得到：

$$I=\mathrm{OSS}\oplus P_I \text{ 和 } Q=\mathrm{OSS}\oplus P_Q \tag{4.15}$$

其中 OSS 表示正交符号扩展结果。

例 4.15 利用公式4.15，I 信道和 Q 信道序列的计算过程如下：

首先与 Walsh 符号输出的第一行模 2 加：

OSS：　01010101 01010101 01010101 01010101 01010101…

P_I：　00000000 00000001 01010010 01110100 01101111…

$I(\oplus)$：01010101 01010100 00000111 00100001 00111010…

与 Walsh 符号输出的第二行模 2 加：

OSS：　10101010 10101010 10101010 10101010 10101010…

$\bar{P}_I$：　11010010 10110101 01010111 01001001 1111…

$I(\oplus)$：01111000 00011111 11111101 11100011 0101…

同样，与 Walsh 符号输出的第一行模 2 加：

OSS：　01010101 01010101 01010101 01010101 01010101…

$\bar{P}_I$：　00000000 00000001 00111101 01110101 10100111…

$Q(\oplus)$：01010101 01010100 01101000 00100000 11110010…

与 Walsh 符号输出的第二行模 2 加：

OSS：　10101010 10101010 10101010 10101010 10101010…

$\bar{P}_I$：　00011101 00110001 00100100 01100110 0011…

$Q(\oplus)$：10110111 10011011 10001110 11001100 1001…

由此得到与 Walsh 符号输出前两行对应的 I 信道和 Q 信道序列。

4.4.6　寻呼信道基带滤波

正交扩频后，I 信道和 Q 信道分别进行基带滤波。其滤波过程与 4.2.3 一节中介绍的步骤一致。基带滤波器的频率响应函数 $S(f)$ 满足图 3.15 的要求。

基带滤波器的归一化频率响应限定是：通带 $0 \leqslant f \leqslant f_p$ 内限定在 $\pm\delta_1$，阻带 $f \geqslant f_s$ 内小于或等于 $-\delta_2$，参数值为：$\delta_1 = 1.5$ dB，$\delta_2 = 40$ dB，$f_p = 590$ kHz 和 $f_s = 740$ kHz。基带滤波后，I 信道和 Q 信道时间符合表 4.2 和表 4.3 的相位关系，如图 4.8 所示。

4.4.7　寻呼信道 QPSK

对寻呼信道的正交相移键控与导频、同步、前向业务信道正交相移键控类似。有关非偏置 QPSK 的问题已在 4.2.4 一节中详细讨论。设 $I(t)$，$Q(t)$ 为基带滤波器的两路正交相输出，$I(t)$，$Q(t)$ 可由 $\theta(t)$ 的相位偏移表示，如表 4.2 所示，因此 QPSK 波形 $s(t)$ 可利用表 4.3 描述。

例 4.16　考虑例4.15的采样数据，I 信道和 Q 信道正交流如下：

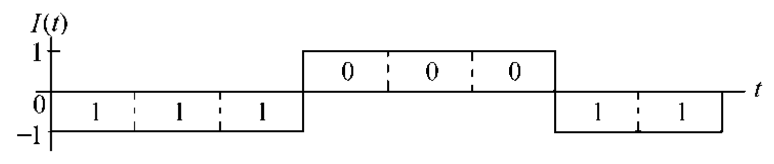
I:	01111000	00011111	11111101	11100011	0101…
Q:	10110111	10011011	10001110	11001100	1001…

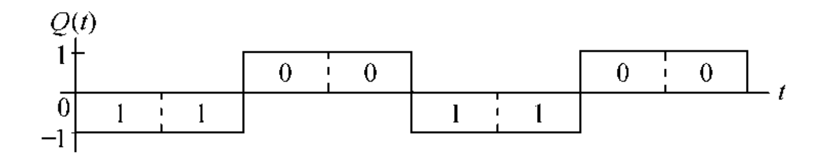
基于两路 BPSK 波形 $s(t)$，即 $I(t) = 11100011$，$Q(t) = 11001100$，可由图 4.22 描述。

寻呼信道(4 800 b/s)

I 信道输出：11100011(−1，−1，−1，　1，　1，　1，−1，−1)

Q 信道输出：11001100(−1，−1，　1，　1，−1，−1，　1，　1)





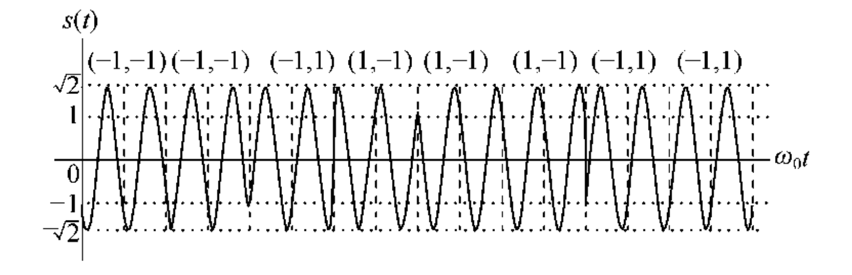


图 4.22　寻呼信道 QPSK 波形 $s(t)$ 图

4.4.8　寻呼信道时隙和消息封装结构

寻呼信道用于向移动台发送非业务信道的消息。

寻呼信道被分割成 80 ms 的时隙，每 80 ms 时隙中又有 4 个长为 20 ms 的寻呼信道帧。2 048 个时隙组成最大的时隙段(20 ms ×2048 = 163.84 秒)，其中的编号从 0 到 2 047。最大时隙段开

始于以 80 ms 为单位的系统时间等于零(模 2 048)时的帧头。

在时隙模式下,移动台利用最大时隙段长的因数的时隙段检测寻呼信道。在非时隙模式下,移动台在所有时刻都检测寻呼信道。移动台在每一指定的时隙都检测寻呼信道,以获得指定时隙的开始位。

20 ms 的寻呼信道帧分成 2 个 10 ms 的长寻呼信道半帧,如图 4.23 所示,半帧的首位称为同步封装标识位(SCI)。

寻呼信道消息封装由寻呼信道消息和填充组成。消息由消息长度、消息体、CRC 组成,填充由 0 或其他要求的比特组成。

基站发送的消息封装可以是同步的或异步的。同步消息封装始于填充半帧的第二位,而非同步消息封装始于前一消息封装结束。

在寻呼信道消息的末尾,基站进行如下工作:

1. 在下一个 SCI 位之前剩下 8 位或更多位时,基站就会在消息后立即发送异步封装消息。在异步寻呼消息封装之后的寻呼消息封装不包括任何填充。
2. 如果下一个 SCI 位之前剩下少于 8 位,基站在消息封装中填充信息,直到下一个 SCI 位出现,而且在 SCI 位之后立即发送同步消息封装。这说明在寻呼信道中,发送的要么是 SCI 位,要么是消息封装的一部分。基站将所有填充位设为 0。

如果 SCI 位之后有消息封装,基站将 SCI 位设为 1,而将其他 SCI 位设为 0,基站发送在同步消息封装中每一时隙开始的第一个消息。允许工作于时隙模式的移动台在激活之后立即获得同步。寻呼信道消息封装结构如图 4.24 所示。

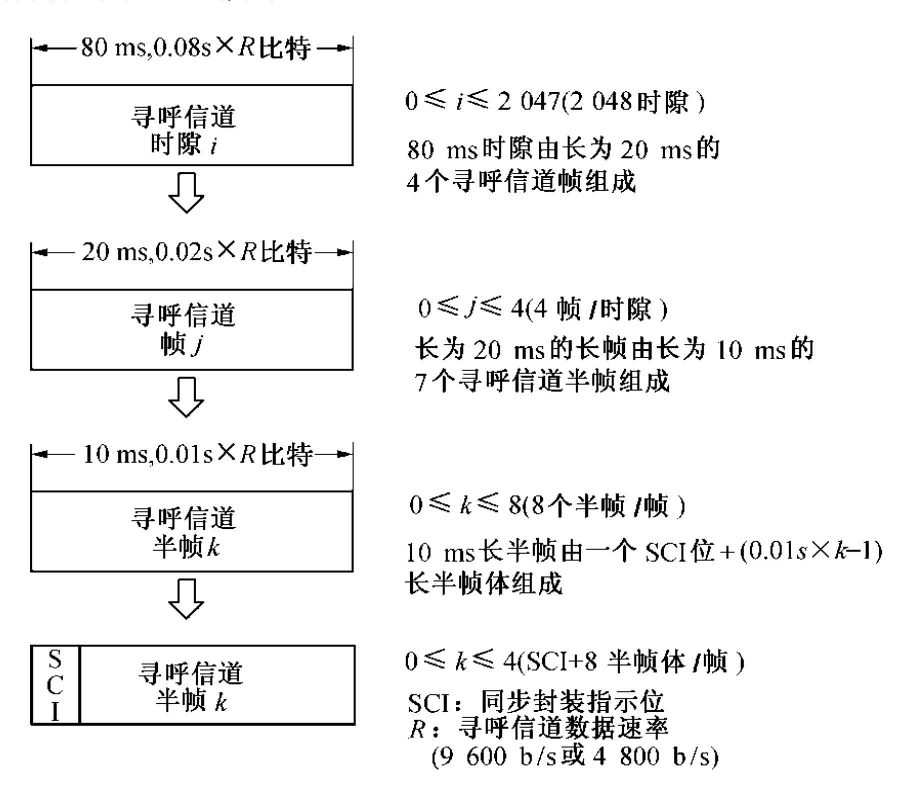


图 4.23 寻呼信道时隙和帧结构

4.4.9 寻呼信道信令消息 CRC

对每一寻呼信道信令消息,其循环冗余编码(CRC)为 30 位。寻呼信道消息由消息长度字段、

消息体和 CRC 字段组成。消息长度字段为 8 位，基站限定最大寻呼消息长度为 148（8 进制）位或 1 184 位。

CRC 生成抽头多项式如下：

$$g(x) = 1 + x + x^2 + x^6 + x^7 + x^8 + x^{11} + x^{12} + x^{13} + x^{15} + x^{20} + x^{21} + x^{29} + x^{30} \tag{4.16}$$

逻辑关系如图 4.15 所示。

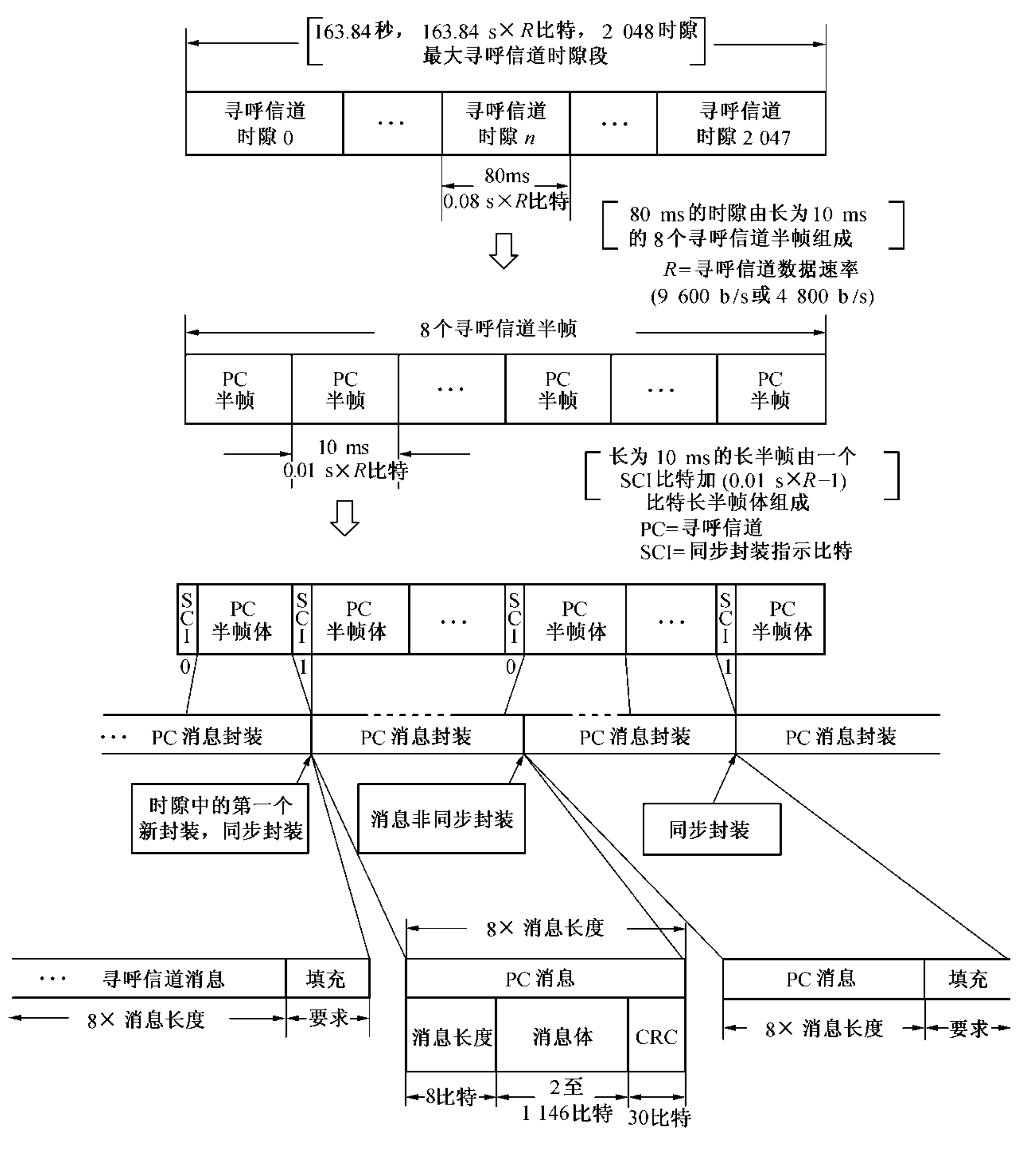


图 4.24　寻呼信道结构例子

CRC 位计算过程如下；逻辑关系如图 4.25 所示。

1. 初始化 LFSR 为 1，目的是即使输入全零也保证 CRC 不为 0。
2. 将开关置下，合上 CRC 编码器反馈路径。

3. k 位输入转换成编码器输出，类似于移位寄存器的反馈路径。
   k 位输入包括 8 位消息长度 + 消息体长度。
4. k 位输出，寄存器锁定 k 次。
5. 将开关置上，打开反馈路径
6. 经过另外 30 次的寄存器锁定，产生 CRC。
7. 寻呼信道消息将以 8 位消息长度、消息体长度、30 位 CRC 的顺序从 CRC 编码器输出。

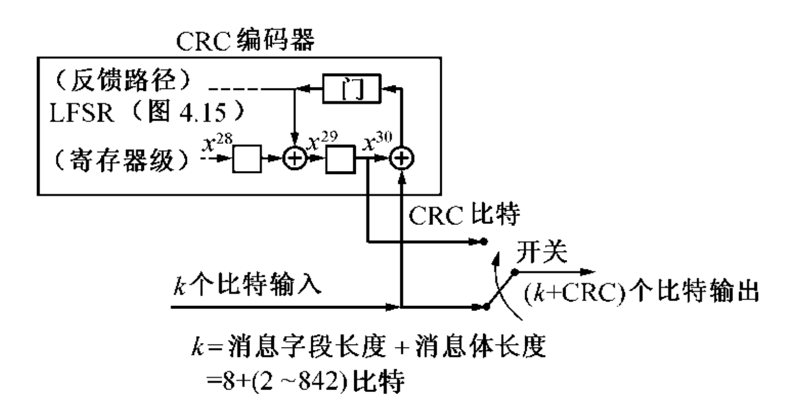


图 4.25　寻呼信道的 CRC 计算图

4.5　前向业务信道(FTC)

前向业务信道用于在呼叫中向特定移动台发送用户数据或信令信息。基站在前向业务信道中发送信息的速率可以是 9 600 b/s,4 800 b/s,2 400 b/s 和 1 200 b/s。前向业务信道帧长 20 ms。

每帧数据速率各自独立。虽然数据速率不同，调制符合速率通过码重复使速率固定在每秒 19 200符号。

作为导频信道的 PN 序列通常也被用做基站前向业务信道的不同偏置的导频 PN 序列。

低速率发送的调制符号发送能量较小。例如，调制符号能量 E_s 如表 4.13 所示，其中 E_b 是信息比特能量。注意，在一个交织块中的符号组成同一帧，因此都会以同样的能量发送。

表 4.13　发送符号能量和数据速率

数据速率	单位调制符号的能量
9 600	$E_s = E_b/2$
4 800	$E_s = E_b/4$
2 400	$E_s = E_b/8$
1 200	$E_s = E_b/16$

以不同速率发送的前向业务信道帧如下：

1. 9 600 b/s 帧，帧长 192 位(20 ms)。
   帧长(192 位) = 信息长度(172 位)+CRC(12 位)+尾比特(8 位)
2. 4 800 b/s 帧，帧长 96 位(20 ms)。
   帧长(96 位) = 信息长度(80 位)+CRC(12 位)+尾比特(8 位)
3. 2 400 b/s 帧，帧长 48 位(20 ms)。
   帧长(48 位) = 信息长度(40 位)+尾比特(8 位)
4. 1 200 b/s 帧，帧长 24 位(20 ms)。

帧长(24 位) = 信息长度(16 位) + 尾比特(8 位)

数据扰码用于前向业务信道中。通过对交织器输出符号和长码 PN 码片中进行模 2 加得到。功率控制子信道信号是连续地在前向业务信道上发送的。该子信道以每 1.25 毫秒 1 个比特的速率(即 800 b/s)发送,见后面的讨论。

前向业务信道的整体结构如图 4.26 所示。前向业务信道的调制参数如表 4.14 所示。

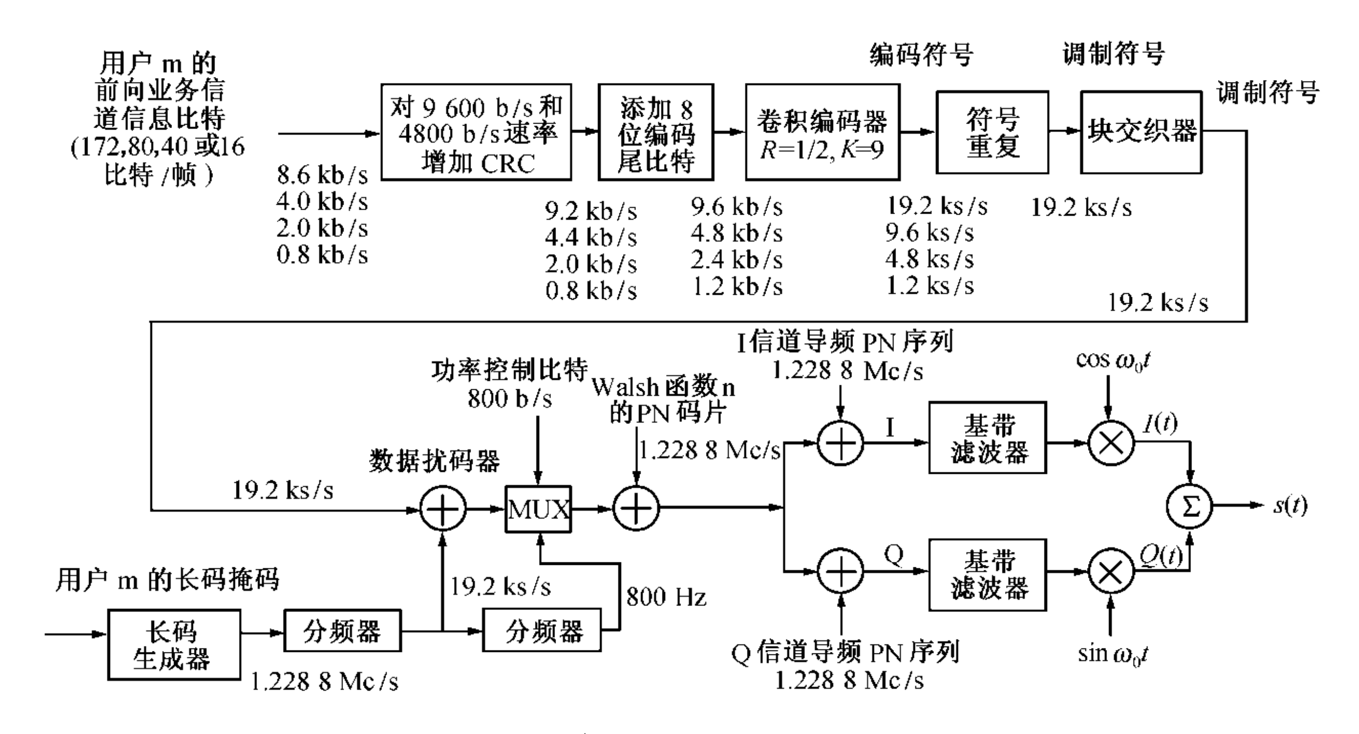


图 4.26　前向业务信道的整体结构

表 4.14　前向业务信道调制参数

参数	数据速率(b/s)				单位
	9 600	4 800	2 400	1 200	
PN 码片速率	1.228 8	1.228 8	1.228 8	1.228 8	Mc/s
编码速率	1/2	1/2	1/2	1/2	比特/编码符号
码重复	1	2	4	8	调制符号/编码符号
调制符号速率	19 200	19 200	19 200	19 200	s/s
PN 码片/调制符号	64	64	64	64	PN 码片/调制符号
PN 码片/比特	128	256	512	1024	PN 码片/比特

4.5.1　前向业务信道的 CRC 计算

每个 9 600 b/s 和 4 800 b/s 均包含 CRC 编码器。CRC 编码器也称为帧质量指示器,但2 400 b/s 和 1 200 b/s 不要求 CRC 计算。对 9 600 b/s 和 4 800 b/s 速率来说,除了 CRC 自身外,还要加入尾比特编码器。9 600 b/s 速率的信号采用 12 位帧质量指示器(CRC)。

4 800 b/s 的速率使用 8 位 CRC,CRC 生成抽头多项式如下

$$g(x) = 1 + x + x^4 + x^8 + x^9 + x^{10} + x^{11} + x^{12} \text{速率为 9.6 kb/s 时} \tag{4.17}$$

和

$$g(x) = 1 + x + x^3 + x^4 + x^7 + x^8 \text{ 速率为 4.8 kb/s 时} \tag{4.18}$$

CRC 计算过程如下,其逻辑如图 4.27 和图 4.28 所示。

1. 所有移位寄存器初始化位设为 1,将开关置上。

2. 寄存器锁住 172 次(192 位帧)或 80 次(96 位帧)并输入信息位(172 位或 80 位)。

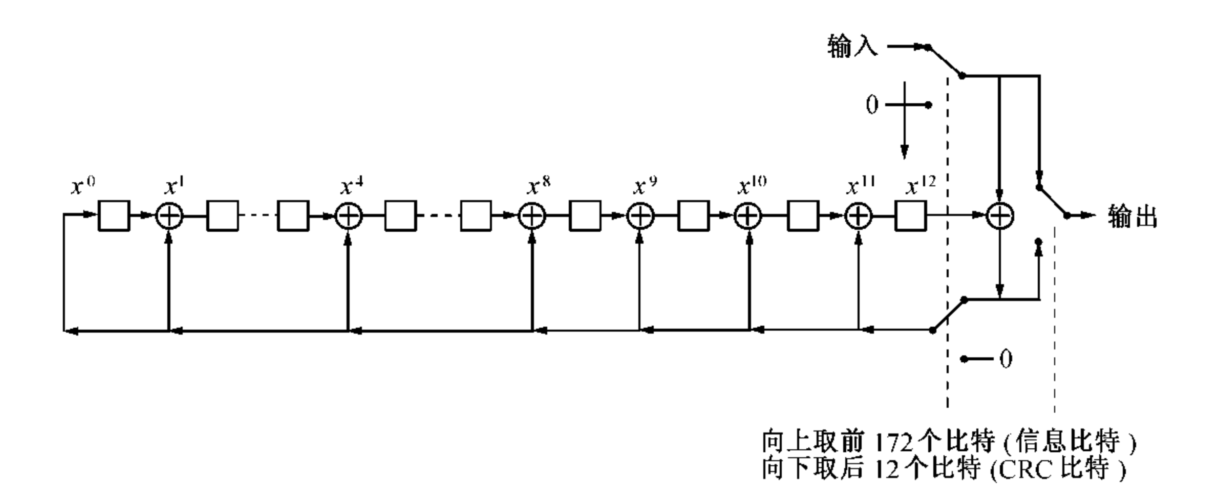


图 4.27 9 600 b/s 速率下前向业务信道 CRC 计算

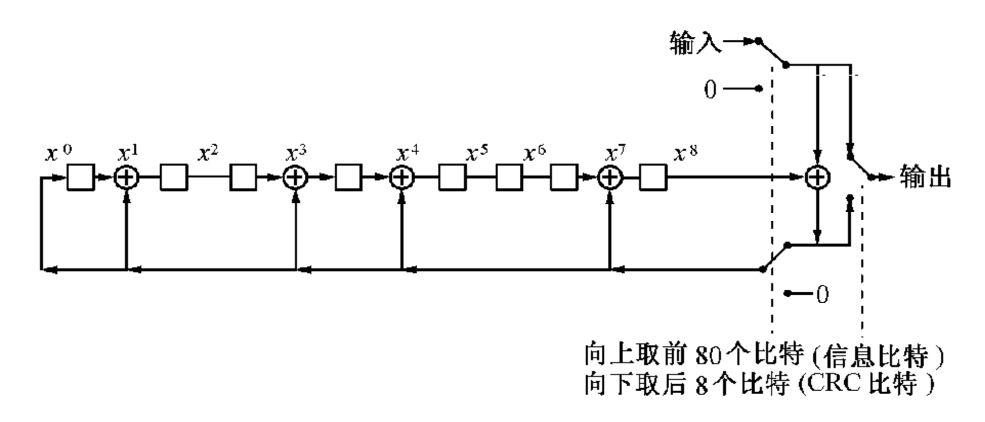


图 4.28 4 800 b/s 速率下前向业务信道 CRC 计算

3. 将开关置下，寄存器锁住另外的 12 次（192 位帧）或 8 次（96 位帧）。另外的 12 位或 8 位输出是 CRC 位。
4. 按计算顺序输出结果。

业务信道帧的后 8 位成为尾比特，都设为 0。

通过增加 CRC 位和尾比特位提高了发送速率。

1. 对 9 600 b/s 帧来说，通过增加 12 位 CRC 和 8 位尾比特，8 600 b/s 的数据速率提高到 9 600 b/s。

   证明：

   每帧 20 ms 有 192 比特，即速率为 9 600 b/s。

   因为速率为 8 600 b/s 时有 172 位信息比特，每比特为 50 b/s。

   那么 12 位 CRC，则有 50 b/s × 12 = 600 b/s。

   从而有 8 600 b/s + 600 b/s = 9200 b/s。又增加 8 位尾比特，得到 50 b/s × 8 = 400 b/s。最后得到 9200 b/s + 400 b/s = 9600 b/s。
2. 对 4 800 b/s 帧来说，通过增加 12 位 CRC 和 8 位尾比特，2 000 b/s 的数据速率提高到 4 800 b/s。

   证明：

   每帧 20 ms 有 96 比特，即速率为 4 800 b/s。

因为速率为 2 000 b/s 时有 80 个信息比特，每个为 50 b/s。

那么 8 位 CRC，则有 50 b/s ×8 = 400 b/s。

从而有 4000 b/s + 400 b/s = 4400 b/s。又增加 8 位尾比特，得到 50 b/s ×8 = 400 b/s。最后得到 4400 b/s + 400 b/s = 4800 b/s。

3. 对 2 400 b/s 帧来说，通过增加 8 位尾比特，2 000 b/s 的数据速率提高到 2 400 b/s。

证明：

每帧 20 ms 有 48 比特，即速率为 2 400 b/s。

每帧 20 ms 有 40 比特，即速率为 2 000 b/s。每比特为 50 b/s。

增加 8 位尾比特，得到 50 b/s ×8 = 400 b/s。最后得到 2000 b/s + 400 b/s = 2400 b/s。

4. 对 1 200 b/s 帧来说，通过增加 8 位尾比特，800 b/s 的数据速率提高到 1 200 b/s。

证明：

每帧 20 ms 有 16 比特，即速率为 800 b/s。因此，每比特为 50 b/s。

增加 8 位尾比特，得到 50 b/s ×8 = 400 b/s。最后得到 800 b/s + 400 b/s = 1 200 b/s。

例 4.17　对4 800 b/s速率的前向业务信道（FTC），由帧内所有位计算得到 CRC，即在 4 800 b/s 速率下每帧 80 个信息比特，利用图 4.28 获得 CRC 的过程如下：

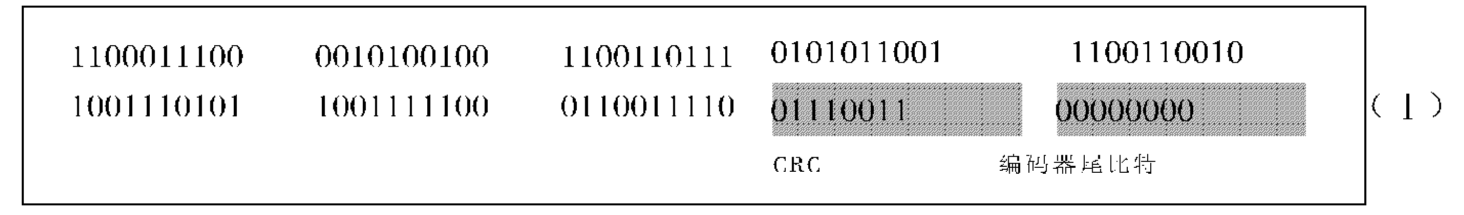
公式 4.18 的生成抽头多项式：g = （11011001）

设定的信息序列：d =（1100011100 0010100100
1100110111 0101011001
1100110010 1001110101
1001111100 0110011110）

寄存器初始状态：I. C. = （11111111）

将 LFSR 设为 1 的目的是：即使输入全为 0，CRC 也能得到非零输出。

输入信息 80 比特，寄存器锁定 80 次（96 比特帧），CRC 为 8 位，获得的 CRC 是 11001110。因此，带 8 位 CRC 加上 8 位全 0 尾比特的生成序列如下：



对于信息速率 9 600 b/s，利用公式 4.17 计算 CRC 留给读者。

4.5.2　FTC 卷积编码

在发送前，FTC 要进行卷积编码，编码率为 1/2，约束长度 K 为 9。

卷积码的生成多项式分别为 $g_1^{(1)}$ = 753（十进制）=（111101011）（二进制）和 $g_1^{(2)}$ = 561 = （101110001）。将生成序列以多项式的形式表示，则有：

$$g_1^{(1)}(x) = 1 + x + x^2 + x^3 + x^5 + x^7 + x^8 \tag{4.19}$$

$$g_1^{(2)}(x) = 1 + x^2 + x^3 + x^4 + x^8 \tag{4.20}$$

利用公式 4.19 与 4.20 的（2，1，8）卷积编码如图 4.12（a）与 4.12（b）所示。因为编码率为1/2，相

应于一个输入位的两个编码符号分别是由 $g_1^{(1)}$ 生成的符号 c_0 和由 $g_1^{(2)}$ 生成的符号 c_1。

例 4.18　利用公式4.19与4.20的 $g_1^{(1)}(x)$ 和 $g_1^{(2)}(x)$ 计算 c_0 和 c_1，编码器的输出公式是

$$c_0 = d_0 + d_1 + d_2 + d_3 + d_5 + d_7 + d_8$$

$$c_1 = d_0 + d_2 + d_3 + d_4 + d_8$$

由图 4.12 及以上输入序列(I)和卷积编码器的全零初始状态，编码器的输出符号是

c_0 =(1000101011 0111011111 1100101011 0010000101 0100010001
1110010001 1011011011 1010011011 0100100100 101101)

c_1 =(1110001010 0111100011 1000111100 0010101100 0100001100
0111001011 0110011101 0100100010 0101100011 100011)

一对串联复接(c_0，c_1)对产生的单路输出符号序列如下：

c =(110101001000110011100011 111101101010111111100000
110111011010000011000100 011100100011000000100101
001010111101001001000111 100111100011110110111001
100001101000111000110001 110000100101110010100111)　(Ⅱ)

这就是卷积编码器的输出符号。

注意，为了适应块卷积表，192 个符号的序列 c 被分成 8 组 24 个符号的形式。

4.5.3　FTC 交织和符号重复

如果信息速率低于 9 600 b/s，前向业务信道的卷积编码符号在块交织之前或之后进行符号重复。速率为 4 800 b/s 时重复 1 次(每个符号连续出现 2 次)，速率为 2 400 b/s 时重复 3 次(每个符号连续出现 4 次)，速率为 1 200 b/s 时重复 7 次(每个符号连续出现 8 次)。但对不同速率(9.6 kb/s，4.8 kb/s，2.4 kb/s 和 1.2 kb/s)，调制符号速率都是 19.2 kb/s。

前向业务信道符号重复后进行块交织。使用的块交织器跨度为 20 ms，也就是在符号速率为 19 200 s/s 时，以 384 个调制码元为一组进行交织。

在表 4.9 到表 4.12 中，给出了对速率为 4.8 kb/s 和 9.6 kb/s 的前向业务信道进行块交织的输入和输出情况。表的读法与同步信道交织一样，都是按列从左到右。表中相同的数字表示符号的重复。

例 4.19　利用表4.10计算速率为 4 800 b/s 的交织器输出。卷积编码器的输出由输入到4 800 b/s 的块交织器的序列 I 给出。由表 4.10，符号序列 I 转换为调制符号序列。按列从左到右所得交织器的输出如下：

111000110010110111110111
010110010001100010000110
010111101000011010111001
010100100110001101100111
101000110101011101100000
111010010111100000101111
100011011001001110100100
010111110100000010001001　(Ⅲ)

表 4. 15 到表 4. 18 可用于计算速率为 2 400 b/s 或 1 200 b/s 的 FTC 交织器输入和输出。

表 4. 15　速率为 **2 400 b/s** 的前向业务信道交织器输入(**TIA/EIA/IS-95**)

1	7	13	19	25	31	37	43	49	55	61	67	73	79	85	91
1	7	13	19	25	31	37	43	49	55	61	67	73	79	85	91
1	7	13	19	25	31	37	43	49	55	61	67	73	79	85	91
1	7	13	19	25	31	37	43	49	55	61	67	73	79	85	91
2	8	14	20	26	32	38	44	50	56	62	68	74	80	86	92
2	8	14	20	26	32	38	44	50	56	62	68	74	80	86	92
2	8	14	20	26	32	38	44	50	56	62	68	74	80	86	92
2	8	14	20	26	32	38	44	50	56	62	68	74	80	86	92
3	9	15	21	27	33	39	45	51	57	63	69	75	81	87	93
3	9	15	21	27	33	39	45	51	57	63	69	75	81	87	93
3	9	15	21	27	33	39	45	51	57	63	69	75	81	87	93
3	9	15	21	27	33	39	45	51	57	63	69	75	81	87	93
4	10	16	22	28	34	40	46	52	58	64	70	76	82	88	94
4	10	16	22	28	34	40	46	52	58	64	70	76	82	88	94
4	10	16	22	28	34	40	46	52	58	64	70	76	82	88	94
4	10	16	22	28	34	40	46	52	58	64	70	76	82	88	94
5	11	17	23	29	35	41	47	53	59	65	71	77	83	89	95
5	11	17	23	29	35	41	47	53	59	65	71	77	83	89	95
5	11	17	23	29	35	41	47	53	59	65	71	77	83	89	95
5	11	17	23	29	35	41	47	53	59	65	71	77	83	89	95
6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96
6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96
6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96
6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96

表 4. 16　速率为 **2 400 b/s** 的前向业务信道交织器输出(**TIA/EIA/IS-95**)

1	3	2	4	1	3	2	4	1	3	2	4	1	3	2	4
17	19	18	20	17	19	18	20	17	19	18	20	17	19	18	20
33	35	34	36	33	35	34	36	33	35	34	36	33	35	34	36
49	51	50	52	49	51	50	52	49	51	50	52	49	51	50	52
65	67	66	68	65	67	66	68	65	67	66	68	65	67	66	68
81	83	82	84	81	83	82	84	81	83	82	84	81	83	82	84
9	11	10	12	9	11	10	12	9	11	10	12	9	11	10	12
25	27	26	28	25	27	26	28	25	27	26	28	25	27	26	28
41	43	42	44	41	43	42	44	41	43	42	44	41	43	42	44
57	59	58	60	57	59	58	60	57	59	58	60	57	59	58	60
73	75	74	76	73	75	74	76	73	75	74	76	73	75	74	76
89	91	90	92	89	91	90	92	89	91	90	92	89	91	90	92
5	7	6	8	5	7	6	8	5	7	6	8	5	7	6	8
21	23	22	24	21	23	22	24	21	23	22	24	21	23	22	24
37	39	38	40	37	39	38	40	37	39	38	40	37	39	38	40
53	55	54	56	53	55	54	56	53	55	54	56	53	55	54	56
69	71	70	72	69	71	70	72	69	71	70	72	69	71	70	72
85	87	86	88	85	87	86	88	85	87	86	88	85	87	86	88
13	15	14	16	13	15	14	16	13	15	14	16	13	15	14	16
29	31	30	32	29	31	30	32	29	31	30	32	29	31	30	32
45	47	46	48	45	47	46	48	45	47	46	48	45	47	46	48
61	63	62	64	61	63	62	64	61	63	62	64	61	63	62	64
77	79	78	80	77	79	78	80	77	79	78	80	77	79	78	80
93	95	94	96	93	95	94	96	93	95	94	96	93	95	94	96

表 4.17　速率为**1 200 b/s** 的前向业务信道交织器输入(**TIA/EIA/IS-95**)

1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46
1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46
1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46
1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46
1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46
1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46
1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46
1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46
2	5	8	11	14	17	20	23	26	29	32	35	38	41	44	47
2	5	8	11	14	17	20	23	26	29	32	35	38	41	44	47
2	5	8	11	14	17	20	23	26	29	32	35	38	41	44	47
2	5	8	11	14	17	20	23	26	29	32	35	38	41	44	47
2	5	8	11	14	17	20	23	26	29	32	35	38	41	44	47
2	5	8	11	14	17	20	23	26	29	32	35	38	41	44	47
2	5	8	11	14	17	20	23	26	29	32	35	38	41	44	47
2	5	8	11	14	17	20	23	26	29	32	35	38	41	44	47
3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48
3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48
3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48
3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48
3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48
3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48
3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48
3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48

表 4.18　速率为**1 200 b/s** 的前向业务信道交织器输出(**TIA/EIA/IS-95**)

1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
9	10	9	10	9	10	9	10	9	10	9	10	9	10	9	10
17	18	17	18	17	18	17	18	17	18	17	18	17	18	17	18
25	26	25	26	25	26	25	26	25	26	25	26	25	26	25	26
33	34	33	34	33	34	33	34	33	34	33	34	33	34	33	34
41	42	41	42	41	42	41	42	41	42	41	42	41	42	41	42
5	6	5	6	5	6	5	6	5	6	5	6	5	6	5	6
13	14	13	14	13	14	13	14	13	14	13	14	13	14	13	14
21	22	21	22	21	22	21	22	21	22	21	22	21	22	21	22
29	30	29	30	29	30	29	30	29	30	29	30	29	30	29	30
37	38	37	38	37	38	37	38	37	38	37	38	37	38	37	38
45	46	45	46	45	46	45	46	45	46	45	46	45	46	45	46
3	4	3	4	3	4	3	4	3	4	3	4	3	4	3	4
11	12	11	12	11	12	11	12	11	12	11	12	11	12	11	12
19	20	19	20	19	20	19	20	19	20	19	20	19	20	19	20
27	28	27	28	27	28	27	28	27	28	27	28	27	28	27	28
35	36	35	36	35	36	35	36	35	36	35	36	35	36	35	36
43	44	43	44	43	44	43	44	43	44	43	44	43	44	43	44
7	8	7	8	7	8	7	8	7	8	7	8	7	8	7	8
15	16	15	16	15	16	15	16	15	16	15	16	15	16	15	16
23	24	23	24	23	24	23	24	23	24	23	24	23	24	23	24
31	32	31	32	31	32	31	32	31	32	31	32	31	32	31	32
39	40	39	40	39	40	39	40	39	40	39	40	39	40	39	40
47	48	47	48	47	48	47	48	47	48	47	48	47	48	47	48

4.5.4　FTC 数据扰码

数据扰码也用于前向业务信道。在调制符号速率为 19.2 ks/s 的条件下，对块交织器输出进行数据扰码。

如图 4.26 所示，通过对块交织器的输出和长码 PN 码片的抽取值进行模 2 加得到扰码。PN 序列的时钟是 1.228 8 MHz，对其进行 64 位抽取首位的结果用于在 19.2 ks/s 条件下对数据进行扰码。长码产生过程将在下面进行描述。FTC 公用长码掩码结构如图 4.29 所示。

长码掩码由 42 位二进制序列组成，作为长码的惟一标识。

FTC 公用长码掩码是 $E_{41} - E_{32}$ = （1100011000）与 $E_{31} - E_0$ = 变序的 ESN 位。

设 ESN 位是（E_{31}，E_{30}，E_{29}，…，E_2，E_1，E_0）。ESN 表示对所有蜂窝系统惟一标识移动台的、由制造商指定的 32 位电子序列号。ESN 的位指配如图 4.30 所示。首次发行时制造商在 32 位 ESN 中 8 个最重要位 E_{31} 到 E_{24} 中指定 MFR 码。E_{23} 到 E_{18} 由 FCC 分配，E_{17} 到 E_0 由制造商惟一指定。因此，变序的 ESN 格式如下。

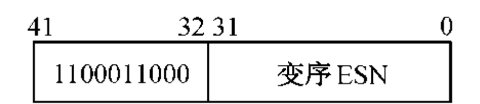


图 4.29　前向业务信道公用长码掩码



图 4.30　ESN 的比特分配

变序 ESN =（E_0，E_{31}，E_{22}，E_{13}，E_4，E_{26}，E_{17}，E_8，E_{30}，E_{21}，E_{12}，E_3，E_{25}，E_{16}，E_7，E_{29}，E_{20}，E_{11}，E_2，E_{24}，E_{15}，E_6，E_{28}，E_{19}，E_{10}，E_1，E_{23}，E_{14}，E_5，E_{27}，E_{18}，E_9）

ESN 改变序列顺序是为了避免长码之间的高度相关性。

长码周期为 $2^{42} - 1$，其寄存器抽头生成多项式 $p(x)$ 如下：

$$p(x) = 1 + x + x^2 + x^3 + x^5 + x^6 + x^7 + x^{10} + x^{16} + x^{17} + x^{19} + x^{21} + x^{22} + x^{25} + x^{26} + x^{27} + x^{31} + x^{33} + x^{35} + x^{42}$$

或　$P = (111101110010000011110110011100010101000000)$　（4.21）

长码的每个 PN 码片由 42 位掩码和生成序列的 42 位状态向量模 2 加运算得到，如图 4.19 所示。

利用如图 4.19 所示的 FTC 长码生成器进行前向业务信道数据扰码，其机制如图 4.31 所示。长码生成器输出的 PN 码片序列工作频率为 1.228 8 MHz。长码分割成 64 位长的单元，64 位的首位被抽取出，用于在 19.2 ks/s 速率下的数据扰码。抽取函数通过 64 取 1 减少了长码大小。

CDMA 系统中的语音保密由用于 PN 扩频的长码掩码的保密性保证。语音保密控制只在业务信道上提供。所有的呼叫均使用 PN 扩频的公用长码掩码来初始化。在确认完成之前不会转向私有长码掩码。

为了开始向私有或公用长码掩码的转换，基站或移动台会在业务信道上发送长码转换请求。

移动台或基站在接收到请求之后会进行转换。基站也可通过发送合适的 PRIVATE-LCN 比特切换方向消息,实现向公用长码掩码的转换。

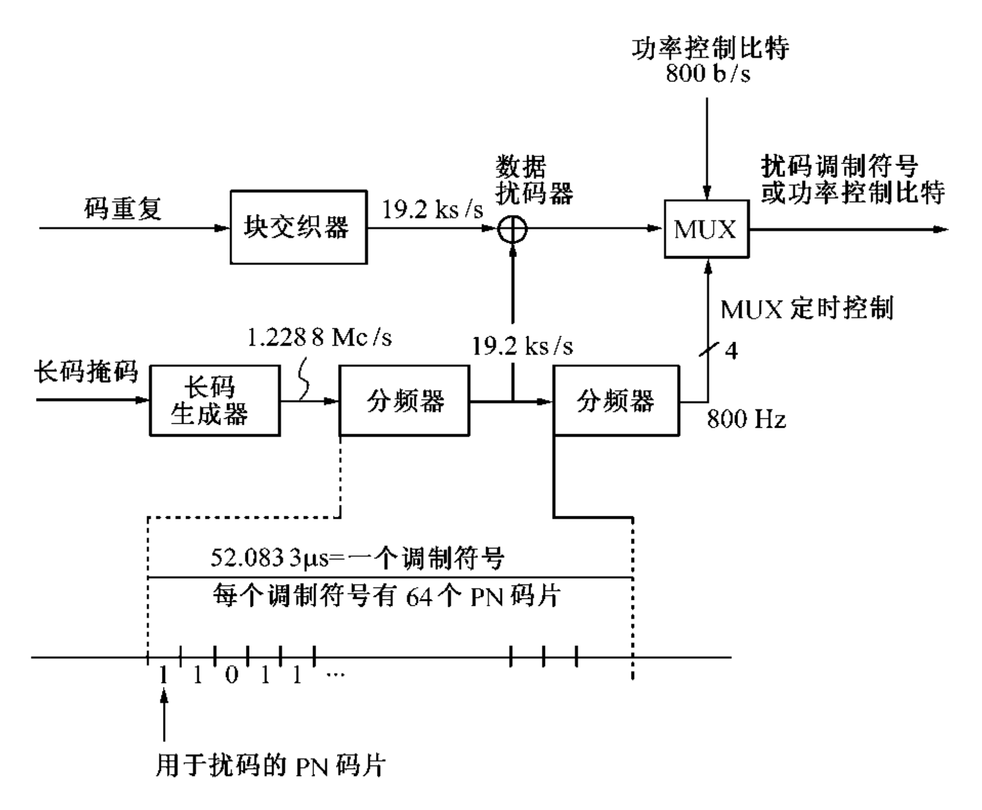


图 4.31 数据扰码机制和定时

例 **4.20** 重新考虑4 800 b/s 速率。参考图 4.31,数据扰码通过对交织器输出序列和长码的抽取值进行异或运算得到。FTC 公用长码掩码由掩码模式 $E_{41}-E_{32}$ =（1100011000）和 $E_{31}-E_{0}$ = 32 位变序 ESN 组成。利用 FTC 公用长码掩码,长码计算如下:

ESN 的位分配由 MFR 码、FCC 分配数、MFR 序列数组成。设

MFR 码: 01001100（8 位）

FCC 数: 101100（6 位）

MFR 序列数: 111010011101100111（18 位）

32 位的 ESN 是（$E_{31}, E_{30}, E_{29}, \cdots, E_{2}, E_{1}, E_{0}$）=（01001100101100111010011101100111）

应用变序 ESN 格式,有:

变序 ESN 位 =（10010111110001001010110011101101）

因此,FTC 公用长码掩码可写为:

公用长码掩码 =（110001100010010111110001001010110011101101）

或长码掩码（LSB-MSB）=（101101110011010100100011111010010001100011）

公式 4.21 的寄存器抽头序列是:

P =（111101110010000011110110011100010101000000）

寄存器的初始内容 =（100000000000000000000000000000000000000000）

利用图 4.19,频率为 1.228 8 MHz 的长码序列生成如下:

1 1011011100110101 0010001111101001 0001100011111011 1010100000100011

2 1000000110001100 0000000101000100 0011100010010010 0110011100101001

3 0011010111100110 1111110001111101 0010010000000100 0011001100110011

4　　1010010010011011 1010100000010100 1101101111000010 1011100110011010

.　　　　.

.　　　　.

.　　　　.

23　　1010101011110111 0000010000101000 1110110001100011 0110010011011110

24　　1010100001000011 1000111101100101 0100101000110110 1101111000100010

.　　　　.

.　　　　.

.　　　　.

观察上面的长码序列，周期为 $2^{42}-1$ 的长码分割成64位长度，而且逐行排列。最左边的一列由每行的首位组成，用来标识抽取值。若将抽取序列分割为每行24位，则得到如下结果：

```
110111100110111110110111
101011111001110111000101
001010100000011100100110   (Ⅳ)
      .        .
      .        .
      .        .
```

然后，对速率为19.2 ks/s的块交织器的输出调制符号和时钟速率在1.228 8 MHz的号码PN码片抽取序列进行数据扰码。

因此，数据扰码由块交织器输出序列(Ⅲ)和抽取二进制序列(Ⅳ)进行模2加得到，即：

数据扰码 = 块交织器输出序列(Ⅲ)$\oplus$抽取二进制序列(Ⅳ)

交织器输出序列(Ⅲ)：　111000110010110111110111 010110010001100010000110…

抽取序列(Ⅳ)：　110111100110111110110111 101011111001110111000101…

扰码数据(Ⅲ$\oplus$Ⅳ)：　001111010100001001000000 111101101000010101000011…

为了对功率控制比特设置开始位置，第一个抽取序列输入到另一个抽取器。有关功率控制比特的内容在以下部分讨论。

4.5.5　功率控制子信道

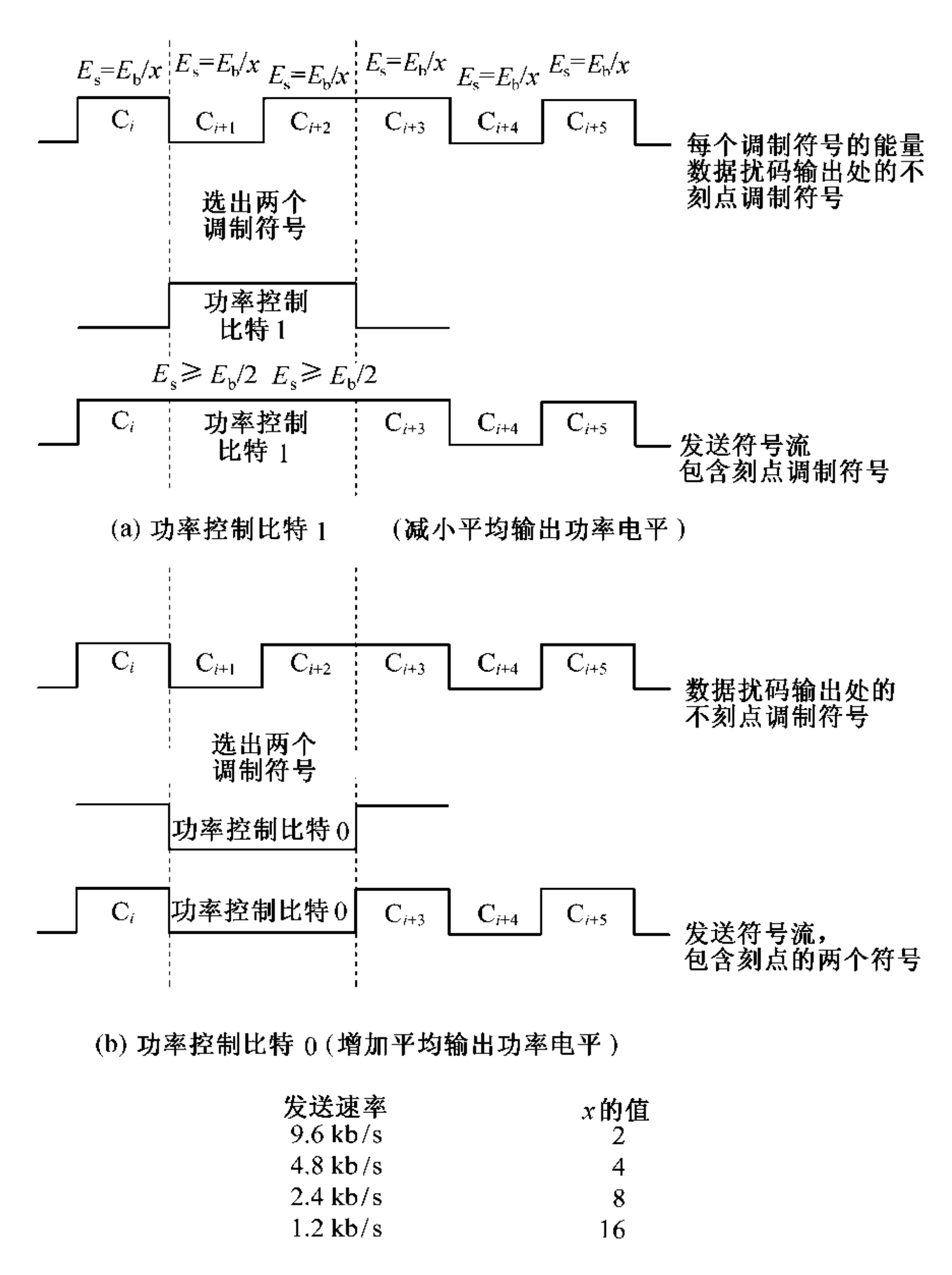
前向业务信道中的功率控制子信道发送功率控制比特的速率是800 b/s，即以每1.25 ms（=1/800）1个比特（0或1）的速率发送。

功率控制比特为0时，指示移动台增加平均输出功率，为1时则减少。移动台根据前向业务信道中收到的有效功率控制比特来调制平均输出功率。如果功率控制比特在移动台发送时隙之后的第二个时隙即1.25 ms时收到，则有效。例如，在反向业务信道(RTC)上功率控制组数为5时接收到信号，则相应的功率控制比特在功率控制组为7(5+2)的业务信道上发送。换句话说，在相应的预测过信号能量的反向业务信道功率控制组之后的二个功率控制组中的前向业务信道上发送功率控制比特。

基站反向业务信道接收机对在1.25 ms期间向所分配的特定移动台的信号强度进行接收和估算，这1.25 ms相当于6(96×1.25/20)个调制符号，基站接收机制用估算来确定功率控制比特值(0

或 1)。基站在相应的前向业务信道上使用截短技术来发送功率控制比特。

一个功率控制比特的长度相当于前向业务信道的两个调制字符(即 104.166 μs = (1.25×10^{-3}/24)×2)。每个功率控制比特取代 2 个连续的前向业务信道调制字符,功率控制比特的发送能量不小于 E_b(每信息比特能量)。如图 4.32 所示,功率控制比特在扰码(图 4.31)之后插入到了 FTC 数据流中。



发送速率	x的值
9.6 kb/s	2
4.8 kb/s	4
2.4 kb/s	8
1.2 kb/s	16

图 4.32 功率控制比特和刻点调制符号

如图 4.33 所示,功率控制比特有 16 个可能的开始位置,每个位置相当于 1.25 ms 期间最初 16 个编码为 0 到 15 的调制字符之一。在每 1.25 ms 期间,来自长码用于扰码的比特总数为 24。这些比特编号从 0 到 23,其中 0 位最先被使用,23 位最后被使用。在图 4.33 中 23,22,21 和 20 这 4 位组成的十进制值(0 到 15)决定功率控制比特的开始位置。其中 23 位是最重要的,20 位最不重要。如图 4.33 所示,4 个二进制值是 0111(十进制值为 14),于是功率控制比特开始位设为 14。图 4.34 指出了速率为 19.2 ks/s 的扰码调制符号和速率为 800 b/s 的收缩功率控制子信道的关系。

例 4.21 讨论前向业务信道扰码之后的 MUX 操作(如图 4.34)。功率控制比特在扰码之后插入

到 FTC 数据流中，由上例得到的扰码数据序列是：

001111010100001001000000

111101101000010101000011

011101001000000110011111

.　　　　　　.

.　　　　　　.

.　　　　　　.

输入到 MUX 中，基站利用收缩技术发送功率控制比特。功率控制比特的长度相当于前向业务信道的两个调制符号。

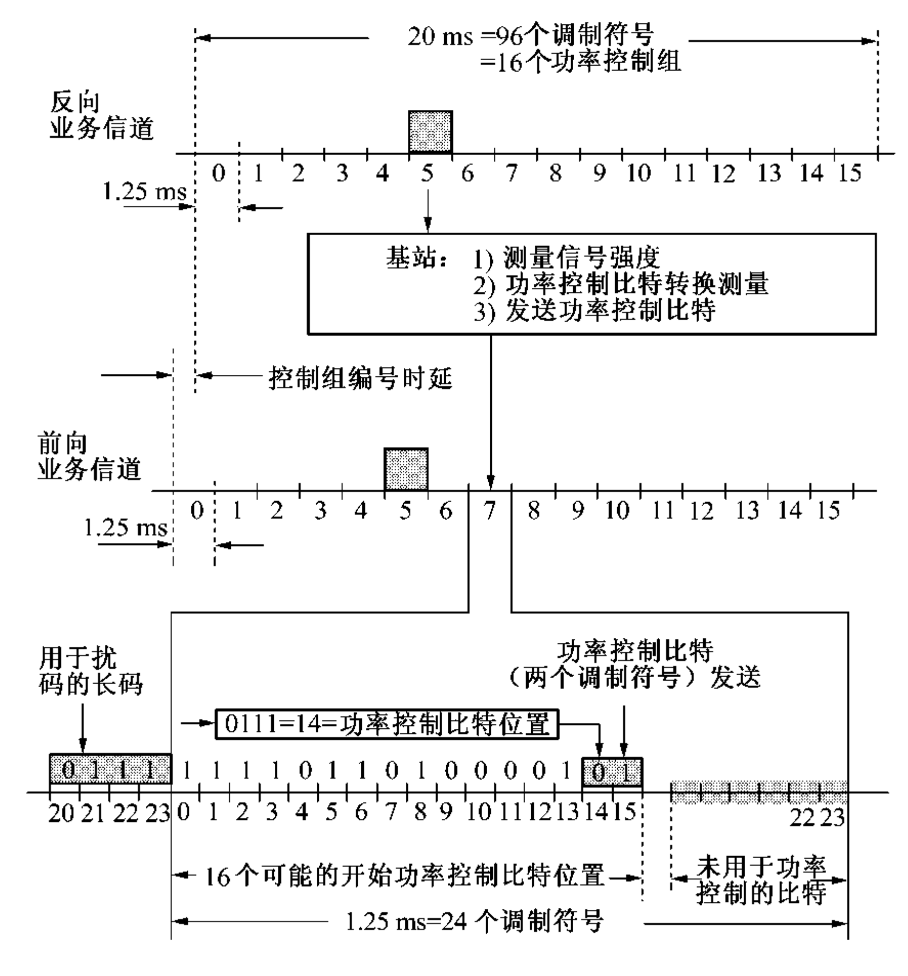


图 4.33　功率控制比特的随机化

抽取长码的前 24 位是 110111100110111110110111，后 4 位即 0111（十进制为 14）决定第二个扰码数据序列（111101101000010101000011）的功率控制比特位置如下：

MUX 输入：	11110110100001 [01] 01000011
MUX 定时控制比特：	0111（十进制为 14）
功率控制比特：	1（假设）
MUX 输出：	11110110100001 [11] 01000011

MUX 输出就是经过扰码的调制符号序列。

通常，编号为 0 的信道（W_0）为导频信道。同步信道编号是 32（W_{32}）。寻呼信道编号是从 1 到 7（W_1，W_2，…，W_7）。其余的 55 个业务信道是 W_8，W_9，…，W_{31}，W_{33}，…，W_{63}。

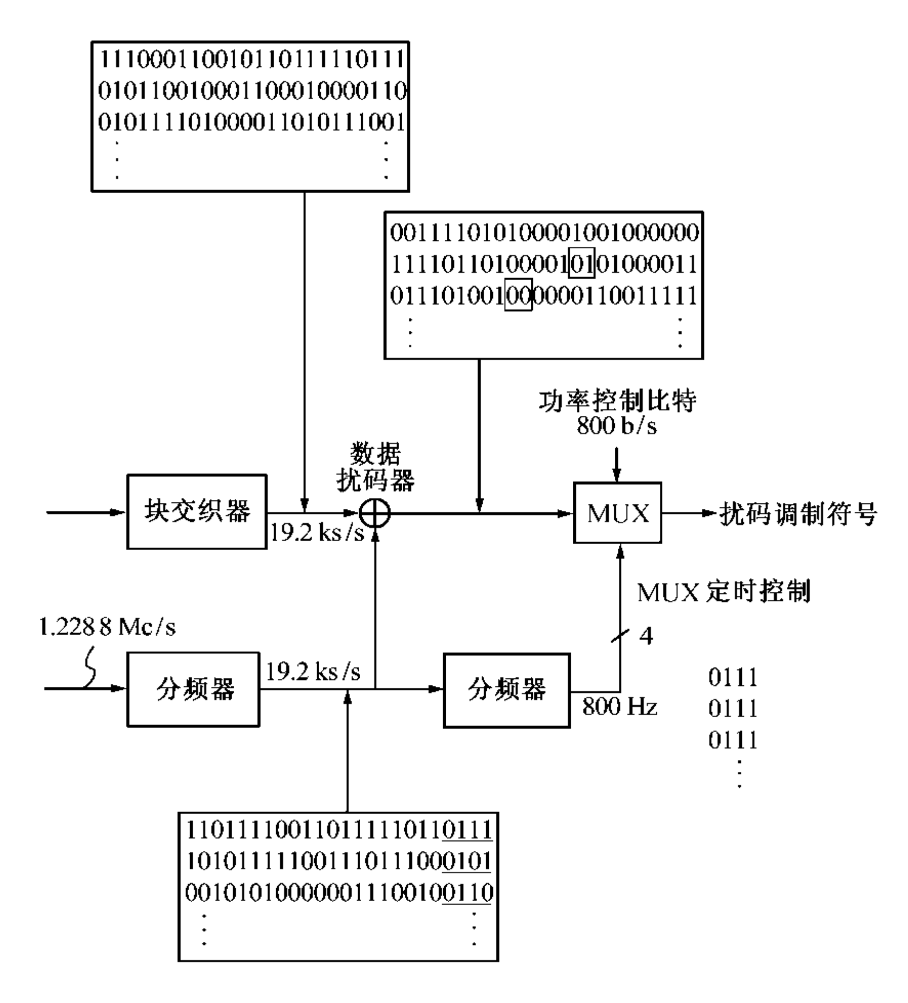


图 4.34 例 4.21 中数据扰码之后的 MUX 操作

4.5.6 FTC 正交相位扩展

在 CDMA 前向链接中，前向业务信道要由 Walsh 函数在固定码片速率 1.228 8 Mc/s 下进行扩展。目的是让信道正交化。扩展所用的 Walsh 函数序号从 0 到 63。在表 4.7 中定义的一个时间正交 Walsh 码分别用于 9.6 kb/s，4.8 kb/s，2.4 kb/s 和 1.2 kb/s。Walsh 码扩展序列周期为 52.083 us(= 64/1.2288 Mc/s)正好等于一个 FTC 调制符号的时间。

例 4.22 讨论 MUX 的多路输出与固定码片速率为 1.228 8 Mc/s 的 Walsh 码 W8 的正交相位扩展。与序号为 8 的 Walsh 函数 W8 对应的 64 位 Walsh 码为：

00000000 11111111 00000000 11111111 00000000 11111111 00000000 11111111 (a)

速率为 19.2 ks/s 的 MUX 输出为

111101101000011101000011 (b)

因为 1.2288 Mc/s/19.2 ks/s = 64 位，多路输出中的一个符号对应于 64 位的 Walsh 码输出，因此将 MUX 输出的第一个符号 1 与 64 位 Walsh 码进行异或：

第一个符号(b) 11111111 … 11111111 … 11111111

Walsh 码(a) 00000000 11111111 00000000 11111111 00000000 11111111 00000000 11111111

(a)⊕(b) 11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000

因此，多路输出序列(b)的前 8 个符号(即 11110110)可扩展到如下 Walsh 码：

11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000

11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000
11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000
11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000
00000000 11111111 00000000 11111111 00000000 11111111 00000000 11111111
11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000
11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000
00000000 11111111 00000000 11111111 00000000 11111111 00000000 11111111

4.5.7　FTC 正交相位扩展和滤波

前向业务信道在正交相位扩展之后由导频 PN 序列进行正交相位扩展，这在 4.1.2 中已经介绍。

I 路和 Q 路导频 PN 序列 $\bar{P}_I$ 和 $\bar{P}_Q$，可由最长的 LFSR 序列 $i(n)$ 和 $q(n)$，$1 \leqslant n \leqslant 32\,767$，生成，LFSR 的初始化值为 100000000000000。

为了得到周期为 2^{15} 的 I 路、Q 路导频 PN 序列，要在 $i(n)$ 和 $q(n)$ 中连续 14 个 0 之后加上 0。在每一个周期都会进行插入，因此导频 PN 序列由连续 15 个 0 输出。

因为导频 PN 序列的速率是 1.228 8 Mc/s，导频序列周期为 26.666（ = 32768/1228800）ms，所以刚好在 2 秒内有 75 个 PN 序列周期。

例 4.23　计算由同相部分 $\bar{P}_I$ 和正交相部分 $\bar{P}_Q$ 组成的正交导频 PN 序列。

为了产生导频 PN 码片，首先要有下面条件：

初始条件：　　　100000000000000

同相抽头向量：　100001011100010

正交相抽头向量：100111100011100

由上面条件，易计算得到 $\bar{P}_I$ 和 $\bar{P}_Q$。

另外，$\bar{P}_I$ 和 $\bar{P}_Q$ 可利用公式 4.5 与 4.6 中的互逆多项式 $i(n)$，$q(n)$ 得到，如下所示：

（I）利用 $i(n)=i(n-15)=0, 1 \leqslant n \leqslant 14, i(15)=1$ 和 $i(n)=\sum_{k=15,10,8,7,6,2} i(n-k)$，$16 \leqslant n \leqslant 32\,767$，计算同相导频 PN 序列 $\bar{P}_I$，先得到 $i(n)$，然后在 14 个连续 0 后面插入 0（如箭头所示）：

↓

$\bar{P}_I$ = 00000000000000010101001001110100011011110011001000
　　00111100001000110100101011010101010111010010011111

$0 \leqslant n \leqslant 99$

（II）与上面类似，利用 $q(n)=q(n-15)=0, 1 \leqslant n \leqslant 14, q(15)=1$ 和 $q(n)=\sum_{k=15,12,11,10,9,5,4,3} q(n-k)$，$16 \leqslant n \leqslant 32\,767$，通过计算 $q(n)$ 得到正交相导频 PN 序列 $\bar{P}_Q$，在 $q(n)$ 的 14 个连续 0 后面插入 0（如箭头所示）：

↓

$\bar{P}_Q$ = 00000000000000010011110101110101101001110001010011
　　10011100011011000111010011000100100100011001100011

$0 \leqslant n \leqslant 99$

例 4.24　对 Walsh 码输出和上例得到的正交导频 PN 序列进行模 2 加实现正交相位扩展。

$$I = \mathrm{WCO} \oplus \bar{P}_I, Q = \mathrm{WCO} \oplus \bar{P}_Q \tag{4.22}$$

其中 WCO 表示 Walsh 码输出。

利用公式 4.22，正交相位扩展（即 I 路序列和 Q 路序列）输出计算如下：

（I）Walsh 码输出与同相 PN 序列 P_I 异或：

WCO：　11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000

P_I：　00000000 00000001 01010010 01110100 01101111 00110010 00001111 00001000

$I(\oplus)$：　11111111 00000001 10101101 01110100 10010000 00110010 11110000 00001000

（II）Walsh 码输出与正交相 PN 序列 P_Q 异或：

WCO：　11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000

$\bar{P}_Q$：　00000000 00000001 00111101 01110101 10100111 00010100 11100111 00011011

$Q(\oplus)$：　11111111 00000001 11000010 01110101 01011000 00010100 00011000 00011011

以上用于 I 流和 Q 流计算的是 Walsh 码输出的第一行。

进行正交相位扩展后，I 流和 Q 流就要输出到 I 路、Q 路基带滤波器中，FTC 滤波器与 4.2.3 中一致。基带滤波器的频率响应 $S(f)$ 满足图 3.15 的要求。

4.5.8　前向业务信道的 QPSK

前向业务信道的 QPSK 与 4.2.4 的一致。设 $I(t)$，$Q(t)$ 为基带滤波器的两路正交二进制数据输出。在无偏置 QPSK 中，两路基带流步调一致，因此在每个 T_b 时间，载波相位发生一次变化。

分别对 $I(t)$，$Q(t)$ 进行余弦和正弦载波的幅度调制得到 QPSK 波形 $s(t)$。

两路正交 BPSK 波形和形成了 QPSK 波形 $s(t)$。

$s(t)$ 表示 QPSK 波形如下：

$$s(t) = I(t)\cos\omega_0 t + Q(t)\sin\omega_0 t$$
$$= \sqrt{2}\cos(\omega_0 t - \theta(t))$$

其中 $I(t) = \sqrt{2}\cos\theta(t)$，$Q(t) = \sqrt{2}\sin\theta(t)$。

$I(t)$，$Q(t)$ 可由表 4.2 中的相位偏移 $\theta(t)$ 表示，QPSK 波形 $s(t)$ 可用表 4.3 描绘。

例 4.25　考虑 I 路和 Q 路 BPSK 数据流的例子。

I：11111111 00000001 10101101 01110100 10010000 00110010 11110000 00001000

Q：11111111 00000001 11000010 01110101 01011000 00010100 00011000 00011011

基于此例中两个 BPSK 脉冲（即 $I(t) = 10010000$ 和 $Q(t) = 01011000$）的 QPSK 波形 $s(t)$ 如图 4.35 所示。

4.5.9　FTC 信道结构和消息结构

基站在业务信道操作时向移动台发送信号消息。在发送前向业务信道消息时，基站把其当做具有某种特定信号业务格式（见表 4.19）的信号业务发送。基站会用一个或多个前向业务信道帧来发送消息。

FTC 帧的第一个信号业务比特是消息开始位（SOM）。如果消息始于该帧，基站将该 2 位设为 1，若消息已在前面帧中开始，则设为 0。基站用剩余信号业务比特来发送 FTC 消息。如果用来发

送消息最后一位的帧包含一些没有使用的信号业务比特，基站将这些位置0填充。

FTC消息由消息长度字段、消息体和CRC字段组成，如图4.36所示。基站的FTC消息结构设置成以8位组表示的长度，消息长度字段占8比特，基站限定消息的最大长度为148(8位组)或1 184比特。

前进业务信道(4 800 b/s)

I信道输出：10010000(-1, 1, 1, -1, 1, 1, 1, 1)

Q信道输出：01011000(1, -1, 1, -1, -1, 1, 1, 1)

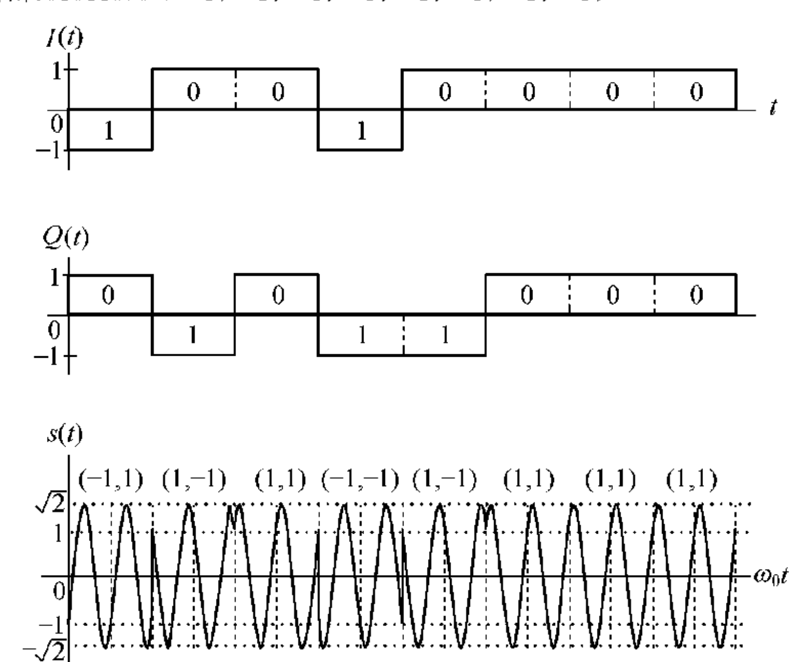


图4.35 前向业务信道QPSK波形 $s(t)$

消息长度	消息体	CRC
8比特	16~1160比特	16比特

图4.36 前向业务信道消息结构

4.5.10 FTC消息CRC字段

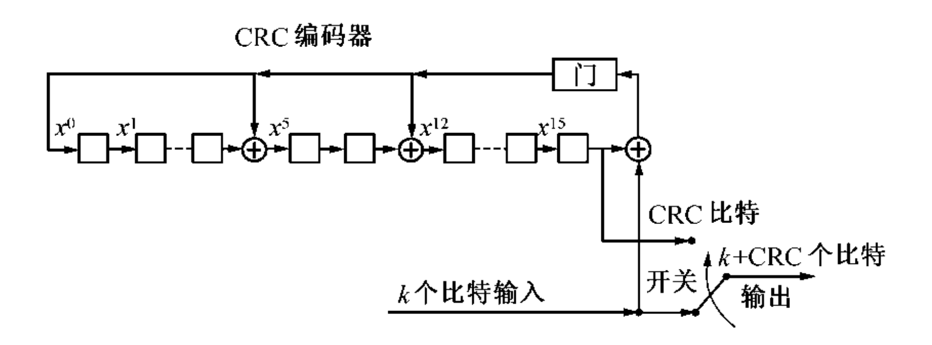
基站为FTC消息设置16比特的CRC字段，CRC计算包括消息长度字段和消息体。

CRC生成多项式见如下所示的标准CRC-CCITT格式。

$$g(x)=1+x^{2}+x^{12}+x^{16} \tag{4.23}$$

CRC值计算过程如下，其逻辑图如图4.37所示。

1. LFSR的初始化置1，目的是对全0输入，CRC值也非0。
2. 将开关置下，关闭CRC编码器的反馈门。
3. k个比特输入进入到编码器的输出以及反馈移位寄存器中。
   k个比特输入由8位消息长度+消息体长度组成。
4. 对于k比特输入过程，寄存器将锁住k次。
5. 将开关置上并打开寄存器的反馈门。
6. 在附加的16次寄存器锁定中，CRC将产生另外16比特输出。时钟脉冲中，产生16位的



k=消息长度字段+消息体长度
=8+(16 至 1160)
以比特为单位

图 4.37　FTC 信令 CRC 计算

CRC 字段。

7. FTC 消息发送的顺序是 8 比特消息长度字段、消息体长度和 16 比特 CRC 字段编码器输出。最后，在 FTC 中发送的信号消息类型总结如表 4.19 所示。

表 4.19　前向业务信道消息

消　息　名	消息类型(二进制)
指令消息	00000001
鉴权查询消息	00000010
信息示警消息	00000011
数据突发消息	00000100
切换导引消息	00000101
模拟切换导引消息	00000110
业务状态系统参数消息	00000111
邻近集更新消息	00001000
发送突发 DTMF 消息	00001001
功率控制参数消息	00001010
重获参数消息	00001011
设置参数消息	00001100
SSD 更新消息	00001101
信息闪动消息	00001110
移动台注册消息	00001111

第 5 章　移动台呼叫处理

呼叫分移动台呼叫和基站呼叫两部分。呼叫处理主要是指移动台和基站之间的消息流协议技术。

本章主要描述基于 TIA/EIA/IS-95 的移动台呼叫。移动台开机之后，进入移动台初始化的系统检测子状态。如图 5.1 所示，呼叫过程由以下几个状态组成：

- 移动台初始化状态——移动台选择和捕获系统。
- 移动台空闲状态——移动台检测寻呼信道的消息。
- 系统接入状态——移动台在接入信道上向基站发送消息。
- 移动台控制业务信道状态——移动台利用前向和反向信道与基站通信。

5.1　移动台初始化状态

在移动台初始化状态中，移动台首先选择一个系统使用，该系统可能是模拟模式或 CDMA 模式。如果是 CDMA 模式的系统，移动台就不断地检测周围各基站发来的导频信号和同步信号。如果是模拟模式的系统，移动台开始模拟模式操作。一旦进入功率增加状态的系统检测子状态，移动台将第一个空闲 ID 的标志置于可操作状态，并将随机确认口令变量置 0。

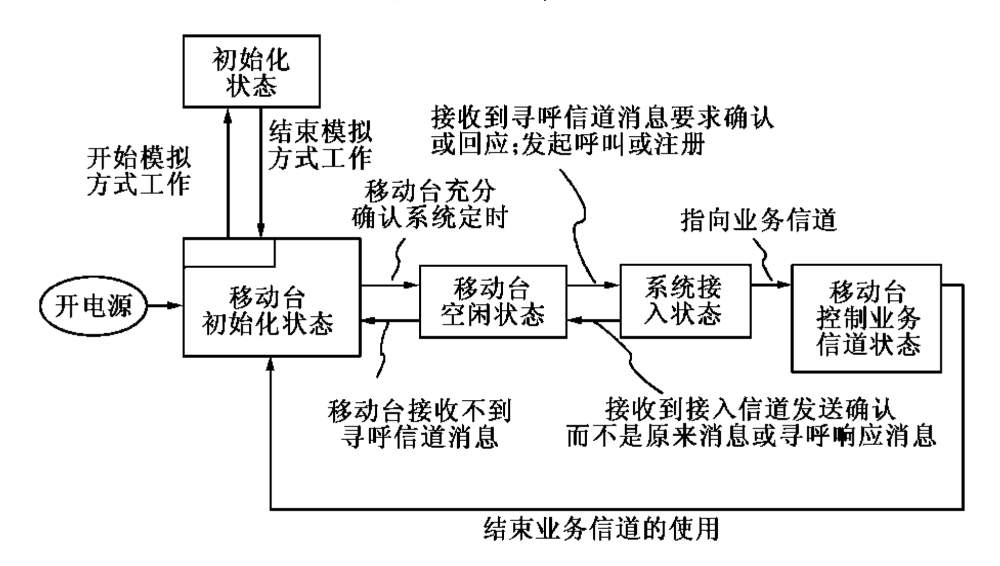


图 5.1　移动台呼叫处理状态

如图 5.2 所示，移动台初始化状态由以下几个子状态组成，即（1）系统检测子状态，（2）导频信道捕获子状态，（3）同步信道捕获子状态以及（4）时间调整子状态。各子状态的作用由下面的表列出。

子　状　态	功　　　能
系统检测	移动台选择可用系统
导频信道捕获	移动台获得 CDMA 系统的导频信道
同步信道捕获	移动台获得系统认证和 CDMA 时钟信息
时钟改变	移动台将自己的时钟同步于 CDMA 系统

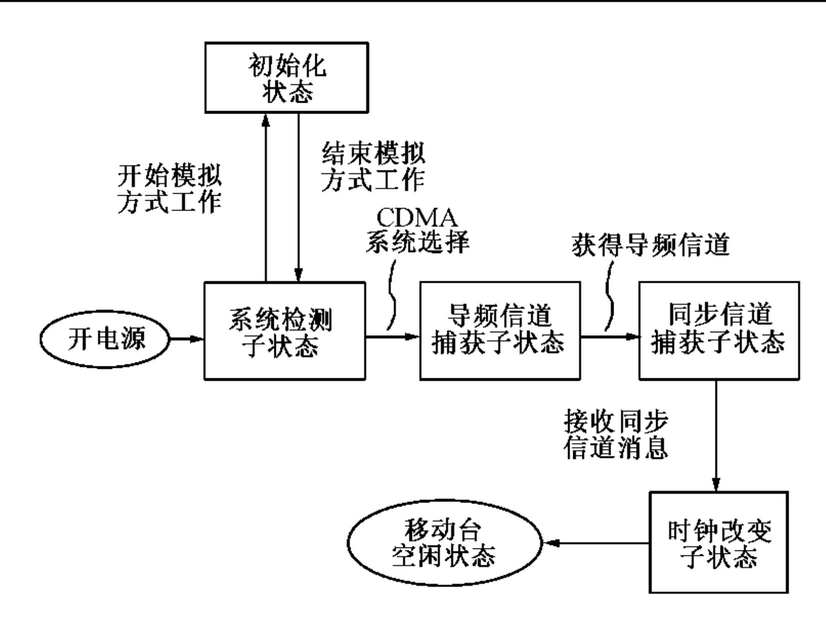


图 5.2 移动台初始化状态

5.1.1 系统检测子状态

在系统检测子状态中,移动台将完成以下功能:

- 移动台检测所用系统。
- 如果使用的是 A(或 B),则将选定的服务系统设定为系统 A。
- 如果移动台使用模拟模式系统,则进入初始化任务。
- 如果移动台使用 CDMA 系统,则先为所选服务系统设置 CDMA 信道序号为第一或第二个 CDMA 信道,接着进入导频信道捕获子状态。
- 如果移动台在捕获 CDMA 系统时在 CDMA 第一个信道失败,移动台会在系统选择过程之前,捕获预备的信道(第一个或第二个)。

5.1.2 导频信道捕获子状态

在导频信道捕获子状态中,移动台为所选 CDMA 系统捕获导频信道. 一旦进入导频信道捕获子状态,移动台调谐到信道序号等于 CDMACHs 的 CDMA 信道上,为导频信道设置编码信道,并寻找导频信道。如果移动台在 T_{20m} = 15 秒内捕获到导频信道,则会进入同步信道捕获子状态。如果在 T_{20m} 没有捕获到,则进入系统检测子状态。T_{20m} 表示了导频信道捕获子状态存在的最长时间。

5.1.3 同步信道捕获子状态

在同步信道捕获子状态中,移动台接收并处理同步信道消息以获取系统设置和定时信息。一旦进入同步信道捕获子状态,移动台为同步信道将编码信道设置成序号为 32。

如果移动台没有在规定的时间 T_{21m} = 1 秒内获取有效同步信道消息,则会进入到系统检测子状态。T_{21m} 表示了接收有效同步信道消息的最长时间。如果移动台虽然在规定的时间 T_{21m} 内获取了有效同步信道消息但移动台支持的服务选择低于基站支持的服务选择,则也会进入到系统检测子状态。如果移动台在规定的时间 T_{21m} 内获取了有效同步信道消息而且移动台支持的协议版本高于或等于基站支持的最低版本,移动台就会储存消息的下列信息:

- 服务选择协议

- 最低服务选择协议
- 网络标识
- 导频 PN 序列偏移
- 长码编码状态
- 系统时间
- 寻呼信道数据速率

移动台可能也会存储消息的下列信息：

- 系统时间开始后占用的闰秒数
- 本地时间相对系统时间的偏移
- 空隙存储时间指示器

然后移动台进入时钟改变子状态。

5.1.4　时钟改变子状态

在时钟改变子状态中，移动台将长码时钟同步于从同步信道消息获得的 CDMA 系统长码时钟，将系统时钟同步于从同步信道消息获得的 CDMA 系统时间当前值的 CDMA 系统时钟。

储存的设定时间等于系统时间，它对应于4路同步信道超帧 320 ms（接收到的同步信道消息的最后 80 ms 超帧之后）减去导频 PN 序列的偏移。也就是说，同步信道消息在时间 320 ms 到消息结束之后的 PN 序列偏移之间有效。对应长码状态与系统长码状态一致，系统长码状态对应在移动台不向外发送的临时值。

5.2　移动台空闲状态

在移动台空闲状态中，移动台检测寻呼信道。可以接收消息、呼入（即终端呼叫）、呼出（即本机呼叫）、注册或发送消息。

一旦进入移动台空闲状态，移动台设置编码信道、寻呼信道数据速率和实施寻呼信道监控。

下面将详细说明寻呼信道监控。

移动台检查接收到的寻呼信道消息的 30 位 CRC 信息。如果移动台在空闲状态工作于非时隙模式，则其一直检测寻呼信道。在 T_{30m} = 3 秒后，无论是否收到寻呼信道的有效消息，移动台都会将计数器复位。T_{30m} 表示收到有效寻呼信道消息的最大时间。

如果在空闲状态工作于时隙模式，移动台在时隙的开始设置 T_{30m} 的计数器来监控寻呼信道，而且 T_{30m} 秒后，无论是否收到寻呼信道的有效消息，移动台都会将计数器复位。如果在移动台监控寻呼信道期间计数器终止，则移动台失去与寻呼信道的联系。在不监控寻呼信道期间，移动台将计数器置为无效。

在移动台处于空闲状态的任何时候，都可以进入移动台初始状态的系统检测子状态。

处于空闲状态时，移动台完成下面过程：

- 寻呼信道检测过程（见 5.2.1）
- 消息确认过程（见 5.2.2）
- 注册过程（见 5.2.3）

- 空闲切换过程(见 5.2.4)
- 开销信息操作响应(见 5.2.5)
- 寻呼匹配操作(见 5.2.6)
- 指令和消息处理操作(见 5.2.7)
- 启动操作(见 5.2.8)
- 消息发送操作(见 5.2.9)
- 关机操作(见 5.2.10)

5.2.1　寻呼信道监视过程

寻呼信道分割成 80 ms 的寻呼信道时隙,在非时隙模式下工作的移动台的寻呼和控制消息可被寻呼信道时隙的任何一个接收。因此非时隙模式要求移动台一直监听寻呼信道。

寻呼信道协议中也规定某个移动台只需在指配的时隙中,发送消息传输进度。

在时隙模式工作时,移动台只需要在某些指定时隙检测寻呼信道。在不检测寻呼信道的时隙里,移动台可以停止或减少功率发送。移动台只有处于空闲状态时才可能工作于时隙模式。也就是说,当移动台在非空闲状态下检测寻呼信道时一定工作于非时隙模式下。工作于时隙模式的移动台在每个时隙段中,在一个周期内,只在其中一个或两个时隙检测寻呼信道。移动台通过注册消息、发送消息或寻呼响应消息中的时隙段序号确定首选时隙段。移动台还可以通过在控制业务信道的终端信息记录中的时隙段序号,确定首选时隙段。时隙段长度 T 为

$$T = 2^i$$

其中 i 是所选时隙段序号。在一个时隙段中有 $16 \times T$ 个时隙。如图 5.3 所示,最小的时隙段含有 16 个长为 80 ms 的时隙,共 1.28 秒。寻呼信道时隙号计算公式如下,要对最大长度的时隙段 12 048 时隙)取模:

$$\text{No.} = \lfloor t/4 \rfloor \bmod 2048$$

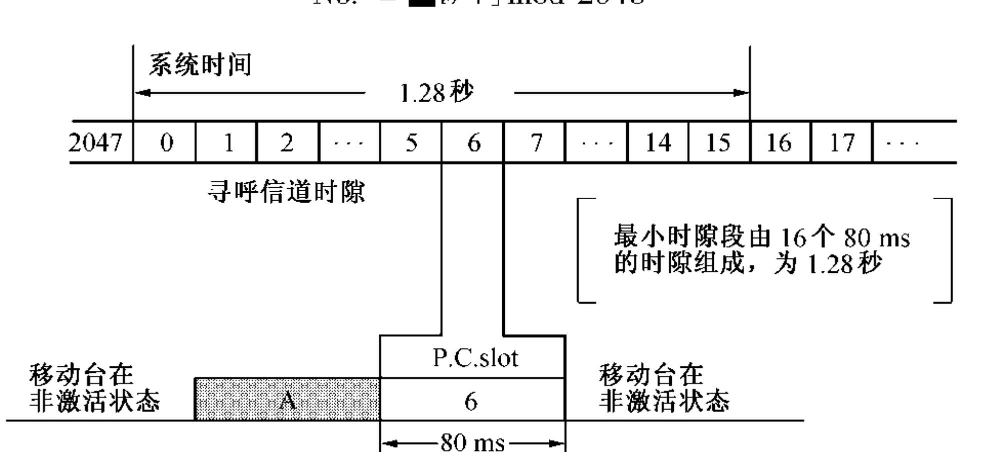


图 5.3　移动台空闲时隙模式结构

其中 t 标识帧中系统时间。如果用户修改了移动台的首选时隙段序号，移动台会修改注册参数。

在所有指定时隙，移动台及时检测寻呼信道以获取指定时隙的第一个比特信息。为了检测指定时隙，移动台利用 Hash 函数从 0 到 2 047 范围中指定时隙。移动台的指定时隙满足 $([t/4] - \text{PGSLOT}) \bmod (16 \times T) = 0$，其中 t 为帧中系统时间，T 是 $T = 2^i$ 指定的最小单位为 1.28 秒的时隙段长度。PGSLOT 由 Hash 函数取得，决定移动台指定的寻呼信道时隙。如图 5.3 所示：一时隙段长为 1.28 秒，计算得到的寻呼信道序号是 6，因此移动台的时隙段开始于序号为 6 的时隙。移动台在时隙 6 开始时检测寻呼信道。16 个时隙之后即时隙号为22(= 6 +16)的时隙是移动台下一个要检测寻呼信道的时隙。

在时隙模式下，移动台要保证其存储的设置参数都是当前的。设置参数由设置消息中获取，即系统参数消息、邻近列表消息和 CDMA 信道列表消息。

与寻呼信道上发送的设置消息相关的是设置消息序号。当设置消息内容改变时，设置消息序号增加。移动台将接收到的设置消息的序号存储。移动台通过检查设置消息的序号来判断其保存的设置参数是否最新。

5.2.2　消息确认过程

确认过程是促进移动台和基站之间的有效消息交换的第二层过程。对从寻呼信道获取的消息的确认在接入信道中发送。移动台通过如下两层来保证确认机制。

- 确认地址类型字段
- 确认序列号字段
- 确认要求字段
- 有效性确认字段

其他消息内容和过程参考第 3 层。

当发送含有确认信息的消息时，移动台将有效确认字段设为 1。对寻呼消息或时隙寻呼消息的确认，移动台将确认序列号字段等于包含移动台识别序号(MIN)的记录的消息序列字段，而且将确认地址类型字段设为 000。

发送没有确认信息的消息时，移动台将有效确认字段设为 0，并令确认地址类型字段和确认序列序号字段分别等于接收的最后一个要求有确认信息的消息的地址类型和消息序列序号字段。如果没有接收到这样的消息，则令确认地址类型字段为 000，确认序列序号字段为 111。

除了对特定消息处理的特殊要求，对应于目的地址的移动台和确认要求字段为 1 的接收消息，移动台发送确认响应。移动台发送的寻呼响应消息包括对每一个寻呼信息记录的确认响应或移动台 MIN 时隙寻呼消息地址。如果一个消息要求响应其他消息指令确认，则响应中要包含确认。否则，移动台确认指令中要包含确认。

如果没有接收到消息指令确认，移动台在发送消息中不必包含确认。如果已经发送了含有确认的消息而且没有接收到要求确认的消息，则移动台在随后发送的消息中不必包含确认。

移动台利用如下原则判断接收的消息是否重复。如果下面都为真，则断定包括移动台地址字段的两消息(或指令记录)重复。

- 接收的消息(字段)：(1)在同一寻呼信道上，(2)时间间隔在 $T_{4m} = 20$ 秒内。T_{4m} 表示移动台

在同一寻呼信道上而且使用同样的序列序号收到两个消息的时间间隔。
- 接收的消息(字段):(1)消息序列序号字段和确认请求字段相同,(2)同样的地址类型字段和地址字段。

如果满足下列条件,移动台断定两个寻呼记录重复。

- 接收的记录:(1)在同一寻呼信道上,(2)所在消息时间间隔在 T_{4m} 秒内或者就在同一消息中。
- 删除接收的记录:(1)消息序列序号字段相同,(2)地址标注同样的 MIN。

移动台将立即删除与前一接收内容重复的消息。

如果满足条件(1)不同基站,(2)不同编码信道,(3)不同 CDMA 信道,则断定为不同的寻呼信道。

如果由以上原则判断的两个消息不是重复,则移动台判断这两个消息不同。

5.2.3 注册及其过程

移动台通过注册通知基站其位置、状态、身份标识、时隙段、移动台类型和其他特征。在移动台被呼叫时,通过注册,基站可以知道移动台的位置,等级和通信能力,确定移动台在寻呼信道的哪个时隙中监听,并能有效地向移动台发起呼叫等。如果工作于时隙模式,移动台提供的时隙段序号参数告知基站移动台检测的时隙。移动台提供的移动台类型和通用空中接口标准(CAI-REV)告知基站移动台容量。

CDMA 系统支持如下不同形式的注册:

1. 开机注册

   开机注册在移动台开机时完成。当移动台开机或从代替服务系统返回或从使用模拟模式返回,移动台会注册。为了防止移动台频繁开关机时的多次注册,移动台会在空闲状态之后和注册之前延迟时间 T_{57m} = 20 秒。T_{57m} 表示开机注册计数器的时限。

2. 关机注册

   关机注册在移动台发出关机指令时完成。当关机注册时,移动台会在注册完成之后才真正关掉电源。如果移动台之前没有在与当前系统标识和网络标识对应的系统中注册,则不会进行关机注册。移动台是蜂窝和网络中的一个单位。网络是系统的子集,系统由系统标识(NID)标注,系统中的网络由网络标识(SID)标注,因此,网络由(SID,NID)惟一确定。NID 值为 0 表示所有不在特定网络中的基站的标识。NID 值为65 535(2^{16} - 1)为移动台的漫游标识以表明 SID 中所有的 NID 都为本地(不是漫游)。

   移动台的本地列表有一个或多个本地(SID,NID)对。如果储存的(SID,NID)对与本地列表中所有(SID,NID)对都不匹配,移动台就处于漫游状态。有两种漫游类型:外来 NID 漫游者,即储存的 SID 与本地列表中某些 SID 相同;外来 SID 漫游者,即储存的 NID 与本地列表中某些 NID 相同。

3. 基于计数器注册

   移动台在计数器截止时注册。基于计数器注册让移动台以固定时间间隔注册。在没有成功进行关机注册时,它允许系统自动注销移动台的注册。基于计数器注册使用寻呼信道时隙计数器。当基站通过系统参数消息的注册周期字段使计数器达到最大值时或称计满,或

称终止，移动台即进行一次注册。基站通过设置注册周期为 0 让基于计数器注册无效。

移动台用如下公式计算计数器截止数

$$E = \lfloor 2^{r/4} \rfloor^{260}$$

其中 E 表示计数器截止数，r 表示注册周期（两次注册之间的间隔）。

在移动台将基于计数器注册指示从 NO 改为 YES 时，将计数器的计数设为在 0 和$E-1$之间的一个伪随机数。由如下线性调和函数生成

$$Z_n = aZ_{n-1} \bmod m$$

其中 $a = 7^5 = 16\ 807$ 且 $m = 2^{31} - 1 = 2\ 147\ 483\ 647$

如果移动台工作于非时隙模式，不论注册计数器标识是否为 YES，移动台会每 80 ms 增加一次基于计数器注册的计数。如果工作于时隙模式，移动台在开始检测寻呼信道时增加基于计数器注册的计数。

4. 基于距离的注册

移动台在当前基站与上次注册基站距离超过门限时注册。移动台通过当前基站与上次注册基站的经度和纬度差的计算结果确定移动距离。如果这个距离超过门限值，移动台注册。

移动台保存基站经度、纬度和用于上次注册的接入信道的基站注册距离。移动台用如下公式计算当前基站与上次注册点的距离。

$$距离 = \left[\frac{(\Delta 纬度)^2 + (\Delta 经度)^2}{16} \right]$$

其中

Δ纬度 $= A - B$

Δ经度 $= (C - D)\cos(\pi/180 \times B/14400)$

A：当前基站经度值，保存在移动台临时存储器中

B：基站经度注册值，保存在移动台半永久存储器中

C：当前基站纬度值，保存在移动台临时存储器中

D：基站纬度注册值，保存在移动台半永久存储器中

5. 基于区域注册

区域是系统和网络中的基站群。通过系统参数消息中注册区域序号字段来标识基站区域指配。移动台进入新区域时注册。如果注册完成，则在列表中增加一个区域，如果计数器到时间，则被删除。在系统接入时，除了由接入成功注册的区域，其他区域的计数器都被激活。在呼叫开始时，所有计数器都被激活。

移动台可以在不同区域注册。通过区域序号加上区域系统标识（SID）以及区域网络标识（NID）来惟一标识区域。移动台保存一个包括所有注册过的区域的列表，列表中每一记录包括区域序号和（SID，NID）对。移动台区域列表能存储 $N_{9m} = 7$ 条记录。N_{9m}表示值为 7 的最大支持区域列表。只有当基站区域注册，（SID，NID）在区域列表记录中能找到才可以认为该基站在区域列表中。移动台在半永久存储器中提供一个记录的存储。

基站通过系统参数消息中的区域总数字段来控制移动台注册的最大区域数。当列表中增加一条记录或者允许的区域总数减少导致记录超过允许数，移动台将从区域列表中移出记录。在区域列表中包括多于允许数的记录时，可根据计数器的值把最早的寿命最长的注册

删除，保证剩下的注册数目不超过允许的数目。

6. 参数改变注册

当如下保存的任何一个参数改变，移动台注册。

- 首选时隙段序号
- 基站类型标志
- 呼叫终端的被激活指示器

移动台的本地列表有一个或多个本地（SID，NID）对。如果保存的（SID，NID）对与列表中非漫游的任何一个（SID，NID）对都不匹配，移动台就处于漫游。

因为注册激活标识并不管理参数改变注册，所以参数改变注册与移动台漫游标识独立。

7. 受命注册

当基站发出请求时移动台注册。通过发送注册请求指令，基站通知移动台注册。在移动台受命和消息处理操作时，实施受命注册。在这一处理操作中，除了开销消息和寻呼消息以外所有消息都被处理。

在移动台指令和消息处理操作时，接收到注册请求指令后马上进行受命注册。在注册请求指令收到 N_{33m} = 0.3 秒之内，移动台进入到带有注册标识的系统接入状态的更新开销信息子状态中。N_{33m}是进入系统接入状态的更新开销信息子状态的最长时间。它是在空闲状态时对所接收消息（除鉴别消息外）的响应。

8. 默认注册

当移动台成功发送起始消息或寻呼应答消息时，基站能借此判断出移动台的位置，不涉及二者之间的任何注册信息交换，这叫做默认注册。在移动台接收到接入信道上发送的注册消息、起始消息或寻呼响应消息的确认之后，注册成功或默认注册。

9. 业务信道注册

一旦基站得到移动台已被分配到业务信道的注册信息时，则基站通知移动台已经注册。

前五种注册形式为一组，称为自主注册，由漫游标识激活。参数改变注册独立于漫游标识。受命注册通过基站的指令消息来初始化。固有注册不包括任何基站和移动台之间的消息交换。当移动台已指配了业务信道，基站通过发送标识请求指令从移动台得到标识消息，从而获得注册信息。通过注册消息通知移动台已经注册。

所有自主注册和参数改变注册都可被激活或禁用。激活的注册形式和相应的注册参数在系统参数消息中获得。

移动台提供了激活或禁用每一计数器的方法。被激活的计数器称为激活计数器。在空闲状态，移动台保持所有激活注册计数器不变或更新。禁用的计数器永远不能过期。在空闲状态时，如果有计数器过期，则移动台保持这一状态以便在该状态下进行其他操作。

移动台提供如下注册计数器：

- 开机初始化计数器

  移动台保持开机初始化计数器。一旦开机初始化计数器是激活计数器，移动台不做注册接入尝试。

- 基于时间的注册计数器

  移动台在计数器过期时注册，一种注册方式是只要计数器到达预定值移动台就注册。计数

器增加每 80 ms 的平均值。当时间增加 80 ms，基于时间的注册计数器使用寻呼时隙计数器来等价于计数器。

- 区域列表记录计数器

  一种自主注册方法是当进入不属于区域列表中的区域时移动台注册。移动台对注册区域列表的每条记录维护一个区域列表纪录计数器。当记录从区域列表中移出，相应的区域列表记录计数器被禁用。计数器的持续时间由区域计数器长度的储存值决定。移动台提供检查激活计数器值的方法，这样就可以计算出列表记录的寿命。

- SID/NID 列表记录计数器

  移动台保持 SID/NID 列表记录的 SID/NID 列表记录计数器。当记录从 SID/NID 列表中移出，相应的 SID/NID 列表记录计数器被禁用。计数器的持续时间由区域计数器长度的储存值决定。移动台提供检查激活计数器值的方法，这样就可以计算出列表记录的寿命。

5.2.4　空闲切换过程

空闲切换发生在移动台处于空闲状态中而且从一个基站的覆盖区移动到另一个基站的覆盖区的情况。当移动台检测到一个足够强的导频信道信号，而该信号并不是当前基站的导频信道信号，移动台就会决定进行空闲切换。

导频信道通过相对零偏置 PN 序列的偏移值来鉴别。导频偏移组成与导频搜索有关的状态组。

下面为移动台处于空闲状态定义如下导频偏移组。

- 激活组

  寻呼信道被监控的前向 CDMA 信道的导频偏移。

- 邻近组

  空闲切换的候选导频信道偏移。其成员在邻近列表消息中指定。

- 剩余组

  在当前 CDMA 频率指配下的系统中，除激活组和邻近组以外的所有可能组。

移动台支持的邻近组至少是 20 个导频。

在空闲状态检测寻呼信道时，移动台不断寻找当前 CDMA 指配的频率上最强的导呼信道信号。

导频搜索必须遵循如下原则：

- 激活组

  激活组中的导频搜索窗大小等于表 5.1 中对应于激活组搜索窗大小的 PN 码片数目（SRCH-WIN-A）。移动台应将激活组的导频搜索窗中心作为导频最先到达的有用多径成分。如果移动台接收到 SRCH-WIN-A 的值大于或等于 13，则会储存并使用 SRCH-WIN-A 中的 13 值。

- 邻近组

  邻近组中的导频搜索窗大小等于表 5.1 中对应于邻近组搜索窗大小的 PN 码片数目（SRCH-WIN-N）。移动台利用其计时器作为计时参考将邻近组的导频搜索窗中心作为导频 PN 序列偏移。

表 5.1　搜索窗尺寸

SRCH-WIN-A SRCH-WIN-N SRCH-WIN-R	搜索窗尺寸 （PN 码片）	SRCH-WIN-A SRCH-WIN-N SRCH-WIN-R	搜索窗尺寸 （PN 码片）
0	4	8	60
1	6	9	80
2	8	10	100
3	10	11	130
4	14	12	160
5	20	13	226
6	28	14	320
7	40	15	452

- 剩余组

  剩余组中的导频搜索窗大小等于表 5.1 中对应于剩余组搜索窗大小的 PN 码片数目（SRCH-WIN-R）。移动台利用其计数器参考定义的计数器将剩余组的导频搜索窗中心作为导频 PN 序列偏移。移动台只搜索导频 PN 序列偏移序号等于导频偏移序号增量的整数倍的剩余组导频。

如果检测到邻近组或剩余组导频信号强于激活组导频信道，移动台会进入如下描述的空闲切换。

在实现空闲切换时，移动台一直工作于非时隙模式，直到在新寻呼信道上接收到至少一个有效消息。之后，移动台会工作于 5.2.1 中讨论的时隙模式。实现空闲切换后，移动台放弃在前一信道上接收的所有未处理的消息。

如果新基站在前基站邻近列表消息之列，移动台用 3 位邻近设置字段决定需要切换到新基站。否则，移动台利用和处理列表中导频一样的过程（将邻近设置字段设为 011）来实现空闲状态切换。

如果邻近设置字段为 000，移动台将接入参数消息序列数设为 NULL 并将导频 PN 序列偏移设置为发送新寻呼信道的基站导频偏移序号。如果移动台没有储存新寻呼信道的设置参数，或者储存的值并不是当前值，移动台将设置消息序列数、系统参数消息序列数以及 CDMA 信道列表消息数设为 NULL，移动台然后用同一编码信道和 CDMA 信道开始监测新基站的寻呼。

假设邻近设置字段为 010，移动台将接入导频序列数设为 NULL，并将导频 PN 序列偏移设置为发送新寻呼信道的基站导频偏移序号。如果移动台设有存储新寻呼信道的设置参数，或者存储的值并不是当前值，移动台将设置消息序列数、系统参数消息序列数以及 CDMA 信道列表消息数设为 NULL。

将当前 CDMA 信道支持的寻呼信道数设为 1。移动台为前一基站调谐到由 CDMA 信道列表消息给出的第一个 CDMA 信道，并开始检测新基站的第一个寻呼信道。如果邻近设置字段为 011，移动台进入系统初始化状态的系统检测子状态。

5.2.5 开销消息操作响应

将寻呼信道信息总结成表 5.2 所示的消息组。组中的消息或者按周期发送或者按需发送。

表 5.2　寻呼信道消息

消　息　名	消息类型(二进制)
系统参数消息	00000001
接入参数消息	00000010
邻近集消息	00000011
CDMA 信道列表消息	00000100
时隙寻呼消息	00000101
寻呼消息	00000110
指令消息	00000111
信道分配消息	00001000
数据突发消息	00001001
鉴权查询消息	00001010
SSD 更新消息	00001011
特征通知消息	00001100
空消息	…

寻呼信道用来发送没有指配给业务信道的控制消息。寻呼信道被分成 80 ms 的时隙。2 048 个时隙组成最大的时隙段(163.84 秒)。在系统时间以 80 ms 为时间单位,模 2 048 为零时,时隙段开始于帧头。工作于时隙模式的移动台,利用长为最大时隙段长度因数的时隙段检测寻呼信道。

寻呼信道上开销消息是:

1. 系统参数消息
2. 接入参数消息
3. 邻近列表消息
4. CDMA 信道列表消息

移动台在接收到开销消息的任何时候都立即进行处理。移动台从接收到的消息数据字段在内部更新储存的信息。

从设置消息和接入参数消息中接收设置参数和接入参数消息。设置消息是:

1. 系统参数消息
2. 邻近列表消息
3. CDMA 信道列表消息

与寻呼信道上发送的设置消息有关的是设置消息序列数。在一个或多个设置消息内容改变时,设置消息序列数增加。对所有接收到的设置消息,移动台储存其所包含的设置消息序列数。同时,移动台储存最近接收到的所有消息包含的设置消息序列数。移动台通过检查所储存的设置消息序列数值来判断其储存的设置参数是否最新。

寻呼消息和时隙寻呼消息也包含设置消息序列数。因此,移动台不必等待设置消息接收,也能判断其储存参数是否过时。在寻呼消息和时隙寻呼消息中,反向参数消息序列数等于当前接入参数消息中发送的序列数。如果移动台接收的寻呼消息和时隙寻呼消息中接入参数消息序列数与其前一个接收储存的接入参数消息序列数不匹配,移动台将接收当前消息并在接入系统之前更新成新参数。

接入参数消息通过接入参数消息序列数建立独立的序列数。接入参数消息序列数通知移动台某一基站发送的接入参数消息是否改变。在接入参数消息中,如果有任何一个字段内容改变,接入

参数消息序列数的值会加 1，移动台储存当前接收的接入参数消息序列数。

通过不同基站、编码信道或者 CDMA 信道发送的寻呼信道都不同。寻呼信道上的设置和接入参数不能用于检测另一不同寻呼信道。如果开销消息的导频 PN 序列偏移序号字段不等于正在检测的寻呼信道的基站的导频偏移序号，则移动台会忽略这些开销消息。

移动台储存从最近检测的寻呼信道上得到的设置参数。移动台在开始检测一个最近被检测过的寻呼信道时，会通过检查设置消息、时隙寻呼消息或寻呼消息中的设置消息序列数储存值決定所储存的参数是否是当前值。

如果如下条件满足，移动台断定所储存的设置参数为当前值。

- 储存的三个设置消息序列数都等于设置消息序列数
- 储存的设置消息序列数不等于 NULL
- 距前一次从寻呼信道上接收有效消息的时间没有超过 N_{31m} = 600 秒。N_{31m} 表示设置参数有效的最大时限。

移动台将特定值 NULL 定义为没有接收到消息序列数或者标识为过时的消息序列数。它不等于任何有效的消息序列数。

如果储存的参数是当前的，移动台就会按照下面的描述处理关于系统参数消息、邻近列表消息和 CDMA 信道列表消息的参数。

1. 系统参数消息

   从寻呼信道上接收到系统参数消息时，将接收的设置消息序列数与系统中储存的移动台参数消息序列数的值进行比较。如果一致，移动台会忽略消息，否则，移动台必须处理 IS-95 中描述的消息的剩余字段。

2. 接入参数消息

   在从寻呼信道上接收到接入参数消息时，将接收的接入参数消息序列数与移动台储存的接入参数消息序列数值进行比较。如果一致，移动台会忽略消息，否则，移动台必须处理 IS-95 中描述的消息的剩余字段。

3. 邻近列表消息

   在从当前寻呼信道上接收到有效邻近列表消息时，将接收的设置消息序列数与储存在移动台邻近列表消息序列数中的值进行比较。如果一致，移动台会忽略消息，否则，移动台必须处理 IS-95 中描述的消息的剩余字段。

4. CDMA 信道列表消息

   在从当前寻呼信道上接收到 CDMA 信道序列消息时，将接收的设置消息序列数与储存在移动台 CDMA 信道列表消息序列数的值进行比较。如果一致，移动台会忽略消息，否则，移动台必须处理 IS-95 中描述的消息的剩余字段。

5.2.6 移动台寻呼匹配操作

寻呼信道上的寻呼消息是寻呼消息和时隙寻呼消息。当接收到寻呼消息时，移动台执行寻呼匹配操作。移动台检查所有消息是否包含移动台标识数(MIN)，如果包含，移动台会在接入信道上发送寻呼消息的寻呼响应消息。

移动台将设置消息序列数的接收值与储存值进行比较。如果比较结果是不匹配，移动台将设

置消息序列数的储存值改成接收值。同时，移动台将接入参数消息序列数的接收值与储存值进行比较。如果比较结果是不匹配，移动台将接入参数消息序列数的储存值设成 NULL。

移动台将自己的 MIN 与每个寻呼消息记录中的 MIN 进行比较。如果记录中的 MIN1 和 MIN2 都是当前的而且 MIN1 和 MIN2 分别与移动台的 MIN1 和 MIN2 都匹配，则寻呼匹配成立。注意：MIN1 代表与指配给移动台的 7 个数字的电话号码对应的 24 位标识数，而 MIN2 代表与指配给移动台的 3 个数字的区号对应的 10 位标识数。如果记录中的 MIN1 是当前值，而 MIN2 不是，MIN1 与移动台的 MIN1 匹配，而且非漫游（SID，NID）与基站的 SID 及 NID 匹配，则寻呼匹配成立。除此之外的任何组合都是非匹配的。

如果寻呼匹配，而且移动台被设置成在漫游状态接收移动台终端呼叫，移动台在寻呼消息接收之后 $N_{33m}=0.3$ 秒之内做出响应则进入系统接入状态的更新开销消息子状态。如果移动台被设置成在漫游状态不接收移动台终端呼叫，则移动台忽略该记录。

5.2.7　移动台指令和消息处理操作

在此过程中，移动台处理除开销消息和寻呼消息以外的所有消息。

移动台将消息的地址字段和相应的移动台标识数据（即 MIN 或 ESN）进行比较。如果匹配，移动台就处理消息，否则移动台就忽略消息。

如下情况下，寻呼信道上接收消息的地址字段与移动台标识数据匹配。

1. 若消息要求确认，而且除了功率周期指令或解锁指令外都不锁住，移动台确认该消息。移动台进入系统接入状态的更新开销信息子状态，除非指定了特定消息，指令或消息的响应时间应在 N_{33m} 秒之内。
2. 若消息不要求确认，移动台就只在消息或指令要求时才发送响应。如果要求响应，移动台进入系统接入状态的更新开销信息子状态，除非指定了特定消息，指令或消息的响应时间应在 N_{33m} 秒之内。
3. 如果是移动台不能处理的消息，移动台会表明拒绝理由来响应拒绝指令。移动台进入系统接入状态的更新开销信息子状态，除非指定了特定消息，指令或消息的响应时间应在 N_{33m} 秒之内。

可以接收 16 个不同的消息或指令。如果消息的任何字段值超过允许范围，移动台会发送移动台拒绝指令，其限定字段值为 00000100（消息字段不在有效范围内）或 00000010（在此状态不能接收该消息）。

5.2.8　移动台发起操作

当用户指定移动台初始化呼叫时，执行移动台发起过程。移动台进入系统接入状态的更新开销信息子状态，其开始在 N_{33m} 秒之内。在系统接入状态，移动台在接入信道上向基站发送消息，在寻呼信道上接收基站消息。

5.2.9　移动台消息发送操作

如果移动台支持移动台消息发送操作，当用户指示移动台发送消息时，移动台消息发送操作被执行。移动台进入系统接入状态的更新开销信息子状态，其消息发送在 N_{33m} 秒之内。是否支持该

操作是任选的。

5.2.10 移动台功率降低操作

当用户指示移动台功率降低时，移动台功率降低操作被执行。移动台更新储存参数并执行其他注册过程。如果不用执行功率降低注册，则移动台直接降低功率。

5.3 系统接入状态

在系统接入状态，移动台在接入信道上向基站发送消息，在寻呼信道上接收基站消息。

如图 5.4 所示，系统接入状态由如下 6 个子状态组成：更新开销消息子状态、移动台起始子状态、寻呼响应子状态、移动台指令或消息响应子状态、注册接入子状态以及移动台消息发送子状态。

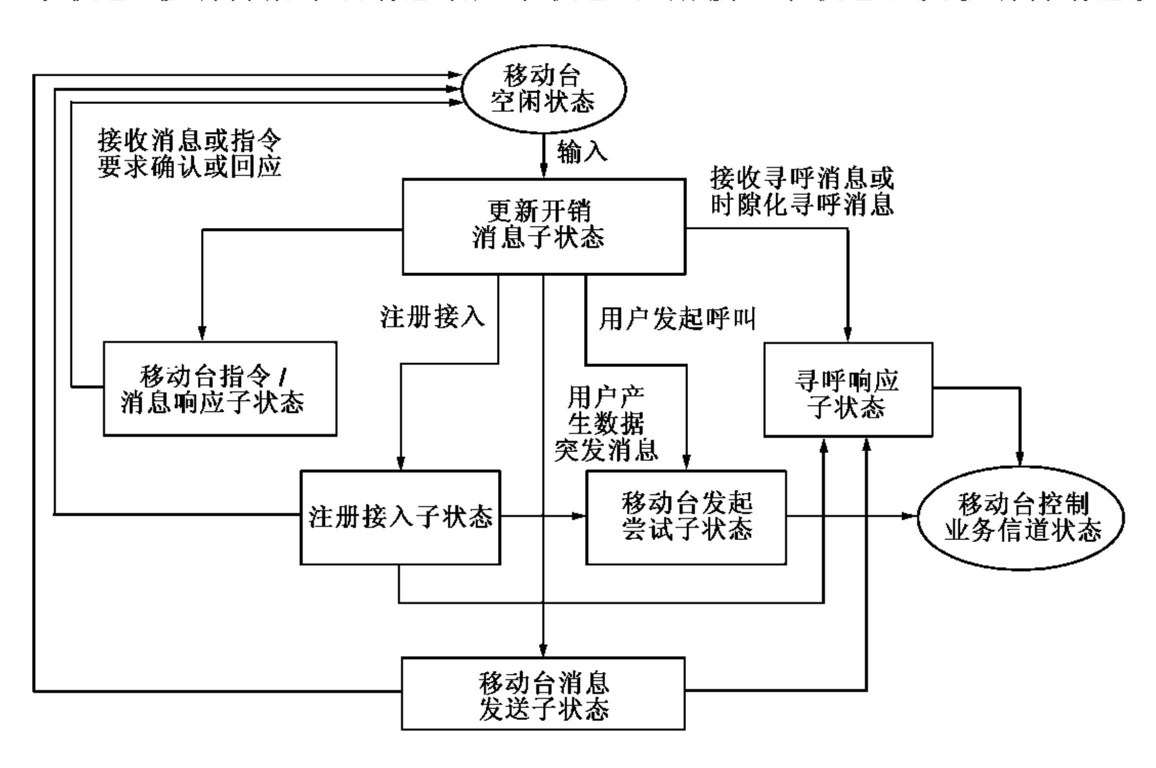


图 5.4 系统接入状态

5.3.1 接入过程

移动台利用随机接入过程在接入信道上发送。随机接入过程的许多参数由基站的接入参数消息提供。发送消息和接收（或接收失败）到消息确认这一整个过程称为接入尝试，如图 5.5 所示。接入尝试中的所有发送称为接入探测。移动台在接入尝试的每个接入探测中发送同样消息。接入探测由接入信道头和接入信道消息封装组成。

在接入尝试中，接入探测组成接入探测序列。接入探测序列由 1 + 接入探测数（所有在同一接入信道上发送的接入探测）组成。接入探测序列中接入探测的最大序号是 1 + 接入探测数，从 1 到 16。移动台发送在接入探测序列中的所有 1 + 接入探测数，除非已经从基站接收到确认。用于接入探测序列的接入信道是从与当前寻呼信道相关的所有接入信道中伪随机选择得到的。接入探测序列的第一个接入探测根据开放环功率电平而指定的功率电平发送。每个接入探测子序列以比前

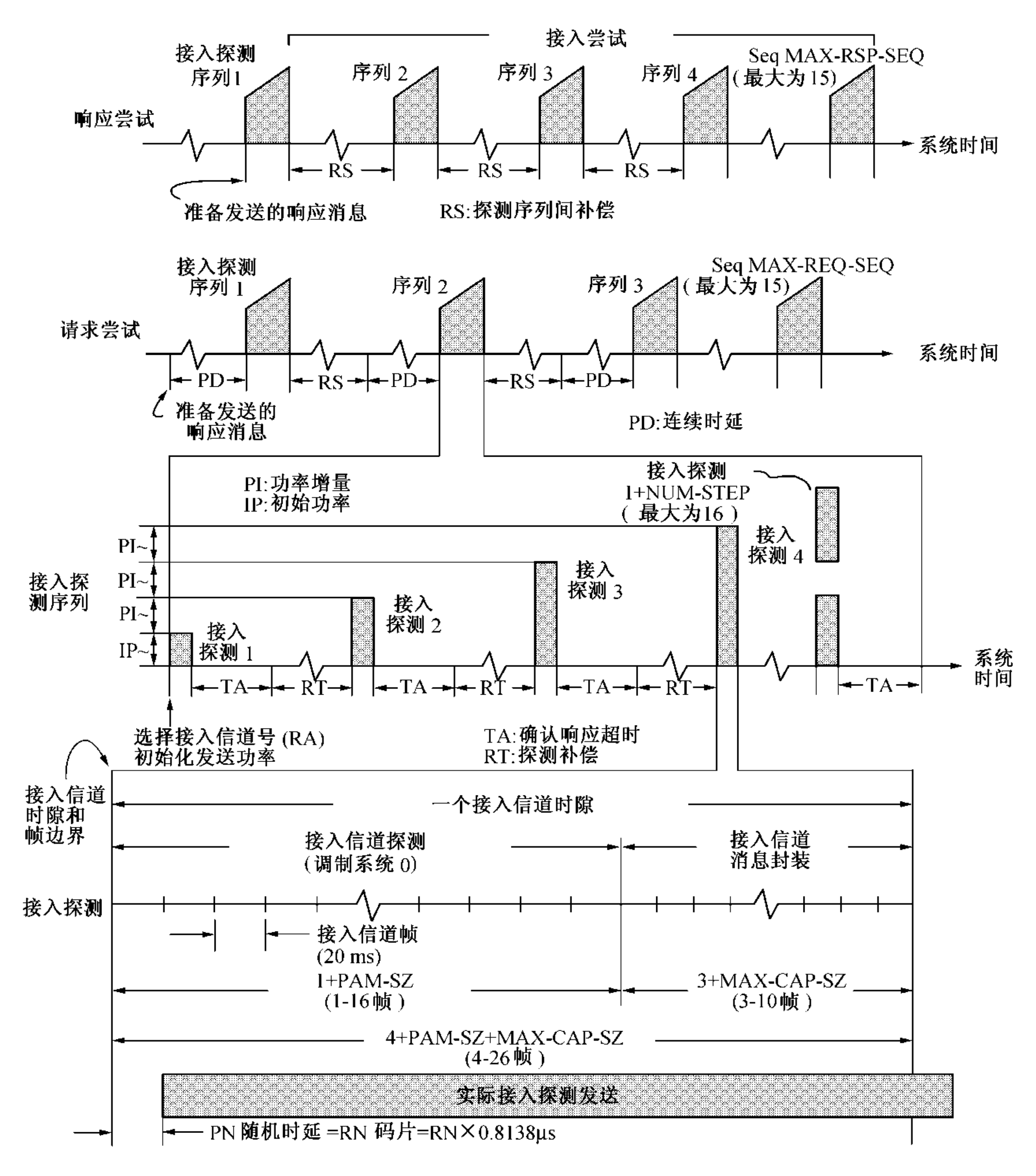


图 5.5　接入信道请求和响应尝试

一接入探测高的功率电平发送。

接入探测和接入探测序列的计数器以接入信道时隙形式表达。接入探测发送开始于接入信道时隙起点。

接入信道上发送的消息有两种类型,即基站消息的响应消息或移动台自动发送的请求消息。对基站消息的响应消息和移动台自动发送的请求消息的处理不同。接入探测序列开始的计数器是伪随机决定的。对接入探测序列,补偿延时[内部探测序列补偿(RS),从 0 到 1 + 接入信道补偿时隙]也是伪随机决定的。对请求接入探测序列,通过持续测试给请求接入探测序列增加延时。持续测试对回复接入尝试不必要,因为基站通过控制其发送消息要求的响应速度直接控制响应消息

的到达速度。移动台对补偿(RS)后的时隙实施随机测试,其参数取决于移动台接入尝试和接入超负荷类型原因。如果测试通过,序列的第一接入探测就从该时隙开始。否则,序列至少被延迟到下一时隙。

接入探测序列的接入探测间的计数器也是伪随机产生。在发送完一个接入探测后,移动台等待特定周期,TA =(2 + 确认超时)×80 ms 是从时隙结尾到从基站接收到确认。如果收到确认,则下一个接入尝试结束。否则,在附加补偿时延(RT)从 0 到 1 + 探测补偿时隙后,发送下一个接入探测。

接入尝试中接入信道发送的精确时间由 PN 随机选择程序决定。对接入尝试,移动台利用依赖于 ESN 的 Hash 函数来计算从 0 到 $2^{TR}-1$ 码片的时延。TR 表示接入信道探测的时间随机化,用于移动台在接入信道探测之前计算 PN 数。移动台将发送时间延迟 RN 个 PN 码片。发送时间调整包括直扩长码延时和正交相位扩展 I 路、Q 路导频 PN 序列延时。这样有效增加了从移动台到基站的覆盖范围。增加了基站对同一接入信道时隙的多个移动台发送信号分别解调的可能性,尤其是多个移动台在基站的一个范围内时。PN 随机化的非随机算法允许基站将 PN 的随机化从来自移动台的实际传播时延分开,从而可以精确预测来自移动台的反向业务信道发送时间。

每次执行接入尝试时,移动台利用前面提到的 Hash 技术计算从 0 到 $2^{TR}-1$ 的数 RN。在接入尝试期间,移动台延迟发送时间,包括用 RN 个 PN 码片的直扩长码和正交相位扩展 I 路、Q 路导频 PN 序列。

移动台实施接入尝试时,发送一个或多个接入探测序列。如果接入尝试是接入信道请求,移动台最多发送接入探测序列最大数;如果接入尝试是接入信道响应,则移动台也最多发送接入探测序列最大数。

发送接入探测序列之前,移动台产生从 0 到当前寻呼信道支持的接入信道数范围的随机数 RA。移动台将该随机数 RA 当做接入探测序列的所有接入探测的接入数。

在发送除第一个接入探测序列外的所有接入探测序列之前,移动台产生从 0 到 1 + 探测序列补偿范围的随机数 RS。移动台将接入探测序列发送延迟 RS 个时隙。

如果接入尝试是接入信道请求,在发送接入序列的第一个接入探测之前,延迟可用的 RS 之后,移动台只有在对时隙的测试通过才在该时隙发送接入序列的第一个接入探测。为了进行连续测试,移动台利用线性拟合技术产生一随机数 RP($0<RP<1$)。当 RP 小于此类接入尝试概率 P 的当前值,测试通过。若 $P=0$,则接入尝试失败而且移动台结束接入尝试,更新注册变量,进入初始化状态的系统检测子状态。

如果接入信道请求是注册,概率 P 计算如下:

$$P=\begin{cases}2^{-PV(n)/4}\cdot 2^{-PM} & \text{如果 } PV(n)\neq 63, n=0,1,\cdots,9\\ 0 & \text{其他情况}\end{cases}$$

$$P=\begin{cases}2^{-PV(n)}\cdot 2^{-PM} & \text{如果 } PV(n)\neq 7, n=10,11,\cdots,15\\ 0 & \text{其他情况}\end{cases}$$

其中 n 是指配给移动台的超负荷类型。$PV(n)$表示移动台超负荷类型的连续值。PM 表示注册接入的连续修正。

如果接入信道请求是消息发送,概率 P 计算如下:

$$P=\begin{cases}2^{-PV(n)/4}\cdot 2^{-PMT} & \text{如果 } PV(n)\neq 63, n=0,1,\cdots,9\\ 0 & \text{其他情况}\end{cases}$$

$$P=\begin{cases}2^{-PI(n)}\cdot 2^{-PMT} & \text{如果 } PV(n)\neq 7, n=10,11,\cdots,15\\ 0 & \text{其他情况}\end{cases}$$

其中 n 是指配给移动台的超负荷类型。$PV(0\sim 9)$表示移动台超负荷类型 0 到 9 的连续值。PMT 表示消息发送的接入信道尝试的连续修正(3 比特)。

如果接入信道请求是除注册和消息发送外的类型,概率 P 计算如下:

$$P=\begin{cases}2^{-PI(n)/4} & \text{如果 } PV(n)\neq 63, n=0,1,\cdots,9\\ 0 & \text{其他情况}\end{cases}$$

$$P=\begin{cases}2^{-PI(n)} & \text{如果 } PV(n)\neq 7, n=10,11,\cdots,15\\ 0 & \text{其他情况}\end{cases}$$

其中 n 是指配给移动台的超负荷类型。$PV(n)$表示移动台超负荷类型的连续值。

连续参数限定了超负荷情况下接入信道请求速率。单独参数 $PV(0\sim 9)$对应于移动台超负荷类型 0~9(正常描述),$PV(10)$对应于移动台超负荷类型 10(测试移动台),$PV(11)$对应于类型 11(紧急移动台),$PV(12)$,$PV(13)$,$PV(14)$和 $PV(15)$预留。移动台利用适当的 PV 参数计算在一个时隙中发送的概率。注册请求发送被发送概率计算中包括的注册连续参数 PM 进一步限定。

移动台接入探测序列的第一个接入探测根据额定开环功率所规定的电平进行发送,其后每个接入探测所用的功率比前一接入探测高一个规定量。在接入探测之间,移动台不发送接入探测序列。

在发送一个接入探测之后,移动台要从时隙末端开始等候确认响应超时,规定时间 TA =(2 + 确认超时)×80 ms。如果接收到确认信息则尝试结束,如果在 TA 内接收不到确认信息,移动台进行如下处理:

1. 如果在该接入探测序列中发送了储存的接入探测数或更少的接入探测,移动台产生一个从 0 到 1 + 接入信道探测补偿范围的随机数 RT(整数探测补偿)。接着移动台会延迟 RT 个附加的接入信道时隙,然后发送下一个接入探测。
2. 如果在该接入尝试中为接入信道响应发送了少于或等于接入探测序列的最大数,移动台会开始另一个探测序列的随机化过程。
3. 除此以外,移动台更新注册变量并进入初始化状态的系统检测子状态。

5.3.2　确认过程

确认过程完成基站和移动台之间消息的可靠交换。移动台利用如下五个字段实现这一机制。字段分别是:确认地址类型、确认序列数、消息序列数、确认要求标识和有效确认标识。这些字段称为 2 层字段,而确认过程也称为 2 层过程。所有其他消息和过程与 3 层有关。所有接入信道消息都有如下四个字段:

1. 确认序列数

   移动台令该字段等于最新接收的要求确认的寻呼信道消息序列数字段值。如果没有接收到满足条件的消息,则将该字段设为 111。
2. 消息序列数

   每一信道使用不同序列数,要求确认消息和不要求确认消息使用不同序列数。寻呼信道和接入信道的消息序列数是 2 比特,前向和反向业务信道消息是 3 比特。

3. 确认要求标识(1 比特)

该字段属于信号消息的 2 层确认字段。如果消息发送者要求确认，则设为 1，否则为 0。

4. 有效确认标识

该字段也属于信号消息的 2 层确认字段。1 表示确认序列数字段包含确认消息的序列数。0 表示确认序列数字段无效。

移动台为接入信道上的消息发送产生一消息序列数设置。开机后，移动台将在接入信道上发送的第一个消息的消息序列数字段设为 000。对每个新的接入尝试即使其内容与前一消息一样，移动台也将消息序列序号加 1 模 8。

在系统接入状态下，移动台检测寻呼信道。当移动台接受到有效确认标识字段值为 1 而且确认序列数字段等于当前发送消息的消息序列序号的消息，移动台判定消息得到确认并结束接入尝试。直到前一个接入尝试结束移动台才开始新的接入尝试。

一旦退出系统接入状态，移动台退出所有正在处理的接入尝试并丢弃相关消息。而在系统接入状态，则消息继续导频搜索，但不再执行空闲切换。

5.3.3 开销子状态更新

在更新开销信息子状态中，移动台一直检测寻呼信道直到接收到当前设置消息。移动台通过比较序列序号来决定是否所有的设置消息已更新。为保证接入参数是最新的，移动台至少接收一个含有接入参数消息序列序号字段的消息，而且如果需要，则等待接入参数消息。

一旦进入更新开销信息子状态，移动台将系统接入状态计数器设成 T_{41m} = 4 秒，T_{41m}是保持寻呼信道上到达的更新开销消息的最长时间。移动台将 PAGES 设为 NO。如果进入更新开销信息子状态时带有寻呼响应指示，移动台将当前接入参数消息序列序号设成接入参数消息序列序号的储存值；否则，将当前接入参数消息序列序号设为 NULL。

在处于更新开销信息子状态时，系统计数器满，移动台将进入初始化状态的系统检测子状态。而在该子状态中，移动台检测寻呼信道。如果移动台宣布失去寻呼信道，则进入空闲状态。如果移动台接收到本地控制指令、锁指令、包含一个与相应移动台标识数据匹配的地址字段的保持要求指令，移动台将所有指令的原因记录在移动台半永久存储器中。对锁指令，移动台应该通知用户锁住状态。移动台进入初始化状态的系统检测子状态，并且直到移动台再次开机或接收到解锁指令，移动台才重新进入系统接入状态。对于保持要求指令，移动台保持解锁条件。移动台应该通知用户保持要求条件。

如果移动台接收到系统参数消息或接入参数消息或邻近列表消息，移动台处理消息中的参数。

如果移动台接收到时隙寻呼消息或寻呼消息，移动台将当前接入参数消息序列序号设置成接收到的接入参数消息序列序号。如果进入子状态时，没有开始或寻呼响应标识，移动台将自己的 MIN 和所有消息记录中的 MIN 进行比较。如果结果匹配，移动台将 PAGES 设为 YES。如果接收到表 5.2 中描述的消息类型的其他消息，移动台会处理消息的所有 2 层字段而忽略其他字段。而其他所有消息都被移动台忽略。

当储存的设置参数是当前值，而且当前接入参数消息序列序号和储存的接入参数消息序列序号相等并都不为 NULL，移动台应使系统进入接入状态并执行如下之一过程。

- 如果 PAGED 等于 YES 或者在进入子状态时有序号响应标识，移动台检测在当前寻呼信道

上是否接收到寻呼匹配的消息。如果接收到，移动台进入寻呼响应子状态，否则移动台进入空闲状态。注意：PAGED 表示在系统接入状态下检测到的寻呼匹配标识。
- 如果进入子状态时带有开始标识，移动台进入开始子状态。
- 如果进入子状态时带有指令/消息响应标识，移动台检测在当前寻呼信道上是否接收到寻呼匹配的消息。如果接收到，移动台进入指令/消息响应子状态，否则移动台丢弃响应并进入空闲状态。
- 如果进入子状态时带有注册标识，移动台进入注册接入子状态。
- 如果进入子状态时带有消息发送标识，移动台进入消息发送子状态。

5.3.4　寻呼响应子状态

在寻呼响应子状态中，移动台发送与来自基站的寻呼消息或时隙寻呼消息对应的寻呼响应消息。如果基站响应寻呼消息时带有鉴别请求，则在该子状态中移动台响应消息。

一旦进入寻呼响应子状态，移动台按照 5.3.1 节描述的过程发送寻呼响应消息。如果消息鉴别有效，移动台利用当前储存的随机查询值计算鉴别模式（2 比特）值和随机查询值（8 比特）。

在寻呼响应子状态中，移动台检测寻呼信道。如果移动台宣布失去寻呼信道，则发送失效并进入空闲状态。如果移动台接收在该子状态下移动台发送的所有消息的确认，移动台结束接入尝试。如果信道指配消息、鉴别查询消息、基站查询设置指令或 SSD 更新消息中不包括确认，移动台将系统接入状态计数器设置为 T_{42m} = 12 秒。T_{42m} 代表收到接入探测的确认后接收 3 层响应的最长时间。

如果寻呼响应消息的接入尝试由接收来自基站的确认结束，移动台更新注册变量。如果在该子状态中，计数器满，移动台进入空闲状态。

如果移动台接收到鉴别查询消息或基站查询设置指令，则移动台按如下过程处理消息：如果移动台在接入尝试过程中接收到消息则忽略。如果移动台在该子状态中对移动台发送的消息进行确认之后接收到消息或者在此消息中包含确认，移动台使系统接入状态计数器失效而且利用 5.3.1 节描述的接入过程来按照惟一查询响应过程中的描述来响应消息。

如果移动台接收信道指配消息，则终止所有处理中的接入尝试。接着按如下过程处理消息：

如果接收的指配模式等于 000，移动台储存前向业务信道的编码信道（储存编码信道 = 接收的编码信道，8 比特），帧偏移（4 比特）给出了前向和反向业务信道帧相对于正常系统时间 1.25 ms（储存的帧偏移 = 接收的帧偏移）的延时，而且消息加密模式标识（储存的消息加密模式 = 接收的消息加密模式），（1）如果设为 000，加密无效，前向和反向信道上发送的所有消息不加密；（2）如果设为 001，标准加密有效，消息加密。如果接收频率包含的标识（1 比特）等于 1，则频率指配（11 比特）（即储存的 CDMA 信道频率指配 = 接收的 CDMA 信道频率指配），移动台进入业务信道上控制状态的业务信道初始化子状态。

如果接收的指配模式等于 001，移动台执行如下操作：如果 CDMA 信道（CDMA 信道频率指配（11 位））列于指配中，移动台设置储存的 CDMA 信道指配等于接收的 CDMA 信道频率指配而且调谐到新的频率指配上。移动台设置储存的接入参数消息序列数（6 比特）为 NULL 而且将导频 PN 序列偏移序号（9 比特）设为接收的所有导频 PN 序列中最强的偏移序号。如果移动台没有为新基站的第一个寻呼信道储存设置参数或者储存值并不是当前值，移动台将储存设置消息序列数（6 比特）、储存系统参数消息序列数、储存邻近列表消息数和储存 CDMA 信道列表消息序列数为 NULL。

接着移动台开始检测被选基站的第一个寻呼信道。如果接收到的模拟控制标识响应(1 比特)等于1,移动台带着寻呼响应重发标识进入更新开销信息子状态。如果接收值为 0,移动台进入空闲状态。如果接收到的指配模式等于 010 而且接收的响应是 1,移动台带着寻呼响应标识进入初始化任务。如果接收到的指配模式等于 010 而且接收的响应是 0,移动台带着等待寻呼标识进入初始化任务。

对锁住指令,移动台使发送失效并在移动台半永久存储器中记录该指令的原因。移动台应该通知用户被锁状态并进入初始化状态的系统检测子状态。直到移动台再次开机或接收到解锁指令,移动台才会重新进入系统接入状态。这一要求优先于其他移动台进入系统接入状态的要求。

移动台在半永久存储器中记录保持必需指令的原因。移动台保持解锁条件并通知用户保持必需的条件。

对于释放指令,移动台进入空闲状态或初始化状态的系统检测子状态。

如果在接入尝试处理过程中接收到 SSD 更新消息,移动台会忽略该消息。如果在确认移动台在该子状态下发送的所有消息之后接收到该消息或者该消息中包括确认,移动台使系统接入状态计数器失效而且按照 5.3.1 中描述的过程来响应该消息。

5.3.5 移动台指令/消息响应子状态

在移动台指令/消息响应子状态中,移动台发送来自基站消息的响应消息。如果基站响应移动台时带有鉴别请求,移动台在该子状态中响应。一旦进入该子状态,移动台按照 5.3.1 中描述的接入过程来发送响应消息。

在移动台指令/消息响应子状态中,移动台检测寻呼信道。如果移动台宣布失去寻呼信道,则使发送失效并进入空闲状态。如果移动台接收了在该子状态下发送消息的确认,则终止接入尝试。如果鉴别查询消息、基站查询确认指令或 SSD 更新消息中没有确认,移动台进入空闲状态。

如果移动台接收到目的地址为移动台的如下消息,移动台会按照下面的描述处理消息:

1. 鉴别查询消息
   如果在接入尝试处理过程中接收到鉴别查询消息,移动台忽略该消息。如果在确认移动台在该子状态下发送的所有消息之后接收到该消息或者该消息中包括确认,移动台利用 5.3.1 中描述的过程,按照惟一查询响应说明来响应消息。
2. 基站查询确认指令
   如果在接入尝试处理过程中接收到鉴别查询消息,移动台忽略该消息。如果在确认移动台在该子状态下发送的所有消息之后接收到该消息或者该消息中包括确认。移动台利用接入过程按照更新共享数据(SSD)说明响应消息。
3. 特征通知消息和本地控制指令
   没有要求
4. 锁指令
   移动台使发送失效并在移动台半永久存储器中记录该指令的原因。移动台应该通知用户被锁状态并进入初始化状态的系统检测子状态。直到移动台再次开机或接收到解锁指令,移动台才会重新进入系统接入状态。这一要求优先于其他移动台进入系统接入状态的要求。
5. 保持要求指令

移动台在半永久存储器中记录保持必需指令的原因。移动台保持解锁条件并将保持要求状态通知用户。

6. SSD 更新消息

如果在接入尝试处理过程中接收到 SSD 更新消息，移动台会忽略该消息。如果在确认移动台在该子状态下发送的所有消息之后接收到该消息或者该消息中包括确认，移动台根据 5.3.1 中描述过程按照更新共享秘密数据（SSD）说明响应消息。

7. 其他消息

如果移动台接收了其他类型，如表 5.1 描述的消息，会处理消息中 2 层的所有字段而忽略其他字段。移动台忽略其他消息。

5.3.6　移动台发起尝试子状态

在移动台发起尝试子状态中，移动台发送开始消息。如果基站响应该消息时带有鉴别请求，移动台在该子状态中响应消息。

一旦进入移动台开始尝试子状态，移动台利用 5.3.1 描述的接入过程发送开始消息。移动台包括不超过消息封装大小尽可能多的发起消息。如果消息鉴别确定，移动台利用当前储存的随机查询值计算鉴别模式（2 比特）值和随机查询值（8 比特）。如果鉴别模式等于 00，鉴别失效。如果鉴别模式等于 01，鉴别数据包括在接入信道消息中适当的位置。随机查询值（0 或 32 比特）的 8 个最重要比特用于鉴别模式的计算（2 比特）。

在该子状态中，移动台检测寻呼信道。如果移动台宣布失去寻呼信道，使发送失效并进入空闲状态。如果接收到该状态中移动台发送消息的确认，移动台终止接入尝试。如果鉴别查询消息、基站查询确认指令或 SSD 更新消息中没有确认，移动台将系统接入状态计数器设为 T_{42m} = 12 秒。

如果开始消息的接入尝试由接收到来自基站的确认结束，移动台将更新注册变量。

如果在此状态中系统接入状态计数器满，移动台进入空闲状态。如果移动台被用户中断呼叫，移动台推出所有正在处理过程中的接入尝试并进入初始化状态的系统检测子状态。

如果接收到如下目的地址为移动台的消息，移动台会按如下所述过程处理消息。

1. 鉴别查询消息

如果在接入尝试处理过程中接收到该消息，移动台忽略消息。如果在确认移动台在该子状态下发送的所有消息之后接收到该消息或者该消息中包括确认，移动台会利用 5.3.1 中描述的过程，按照惟一查询响应说明来响应消息。

2. 基站查询确认指令

如果在接入尝试处理过程中接收到 SSD 更新消息，移动台会忽略该消息。如果在确认移动台在该子状态下发送的所有消息之后接收到该消息或者该消息中包括确认，移动台根据 5.3.1 中描述的过程按照更新共享秘密数据（SSD）说明响应消息。

3. 信道指配消息

移动台终止所有在处理过程中的接入尝试。按如下过程处理消息：

如果接收的指配模式（3 比特）等于 000，移动台储存前向业务信道的编码信道（储存编码信道 = 接收编码信道）、帧偏移（储存帧偏移 = 接收帧偏移，4 比特）、消息加密模式标识（储存消息加密模式 = 接收消息加密模式）以及频率包含标识为 1 时的频率指配（储存 CDMA 信道 = 接收 CDMA 信道频率指配）。而且移动台进入业务信道上控制状态的业

务信道初始化子状态。

如果接收的指配模式等于001，移动台执行如下操作：如果CDMA信道（CDMA信道频率指配）列于指配中，移动台设置储存的CDMA信道指配 = 接收的CDMA信道频率指配而且调谐到新的频率指配上。移动台设置储存的接入参数消息序列数（6比特）为NULL而且将导频PN序列偏移序号（9比特）设为列表中最强导频PN序列的偏移序号（接收的导频PN序列偏移序号）。如果移动台没有为新基站的第一个寻呼信道储存设置参数或者储存值并不是当前值，移动台将确认消息序列数（6比特）、系统参数消息序列数、邻近列表消息数和信道列表消息序列数设为NULL。接着移动台开始检测被选基站的第一个寻呼信道。如果接收到的模拟控制信道标识响应（1比特）等于1，移动台带着起始标识进入更新开销信息子状态。

如果接收到的指配模式等于010而且接收到的模拟控制信道标识响应是1，移动台带着起始标识进入初始化任务。

如果接收到的指配模式等于011，移动台储存系统标识、语音信号衰减编码、语音信道数、管理音频音调色彩编码和消息加密模式标识，而且带着开始标识进入设置初始语音信道任务。

4. 特征通知消息

如果接收的释放指令等于1，移动台终止所有正在处理过程中的接入尝试并进入空闲状态或初始化状态的系统检测子状态。如果接收的释放指令等于0，移动台将系统接入计数器复位到 T_{42m} = 12秒。

5. 插入指令

移动台终止所有在处理过程中的接入尝试并进入空闲状态。

6. 本地控制指令

没有要求

7. 锁指令

移动台使发送失效并在移动台半永久存储器中记录该指令的原因。移动台应该通知用户被锁状态并进入初始化状态的系统检测子状态。直到移动台再次开机或接收到解锁指令，移动台才会重新进入系统接入状态。这一要求优先于其他移动台进入系统接入状态的要求。

8. 保持要求指令

移动台在半永久存储器中记录该指令的原因。移动台保持解锁状态并通知用户保持要求状态。

9. 释放指令

移动台进入空闲状态或初始化状态的系统检测子状态。

10. 记录指令

移动台终止所有在处理过程中的接入尝试并进入空闲状态。

11. SSD更新消息

如果在接入尝试处理过程中接收到SSD更新消息，移动台会忽略该消息。如果在确认移动台在该子状态下发送的所有消息之后接收到该消息或者该消息中包括确认，移动台使系统接入状态计数器失效而且利用5.3.1中描述的过程并按更新共享秘密数据（SSD）说

明响应消息并使系统接入状态计数器终止。

12. 其他消息

如果接收到表 5.2 中描述的消息类型的其他消息，移动台会处理消息的所有 2 层字段而忽略其他字段。而其他所有消息都被移动台忽略。

5.3.7　注册接入子状态

在注册接入子状态中，移动台向基站发送注册消息。如果基站回复带有鉴别请求，移动台就在该子状态中响应。一旦进入注册接入子状态，移动台就按照 5.3.1 中描述的接入过程发送注册消息。如果启用消息鉴别，移动台利用当前储存的随机查询值（0 或 32 比特）计算鉴别数据值（18 比特）和随机查询值（8 比特）。

在该子状态中，移动台检测寻呼信道。如果移动台宣布失去寻呼信道，则发送失效并进入空闲状态。如果移动台接收在该子状态下移动台发送的所有消息的确认，移动台结束接入尝试。如果鉴别查询消息、基站查询设置指令、SSD 更新消息、释放指令或由用户关机引起的注册指令中不包括确认，移动台将做如下之一的工作。

- 移动台更新注册变量并关机。
- 否则，移动台进入空闲状态。

如果注册消息的接入尝试由于接收到基站确认结束，移动台更新注册变量。

如果接收用户的发起呼叫指令，移动台放弃所有正在处理过程中的接入尝试并进入移动台开始尝试子状态。

如果接收寻呼消息或时隙寻呼消息，移动台将自身 MIN 和消息中所有记录的 MIN 进行比较。如果匹配，移动台放弃所有正在处理过程中的接入尝试并进入寻呼响应子状态。

如果接收到其他所有目的地址为移动台的消息，移动台会按照描述处理消息。

1. 鉴别查询消息
2. 基站查询确认指令

如果在接入尝试处理过程中移动台接收以上两消息中的任何一个，或由于关机初始化注册接收，则移动台忽略消息。如果移动台在该子状态中对移动台发送的消息进行确认之后接收到其中一消息或者其中一消息中包含确认，移动台将利用 5.3.1 节描述的接入过程，在鉴别查询消息的惟一查询响应过程中和基站查询确认指令的更新共享秘密数据（SSD）过程中响应消息。

3. 特征通知消息

没有要求

4. 本地控制指令

没有要求

5. 锁住指令

移动台使发送失效并在移动台半永久存储器中记录该指令的原因。移动台应该通知用户被锁状态并进入初始化状态的系统检测子状态。直到移动台再次开机或接收到解锁指令，移动台才会重新进入系统接入状态。这一要求优先于其他移动台进入系统接入状态的要求。

6. 保持要求指令

   移动台在半永久存储器中记录该指令的原因。移动台保持解锁状态并通知用户保持要求状态。

7. 注册接收指令

   没有要求

8. 注册拒绝指令

   没有要求

9. 释放指令

   移动台进入空闲状态或初始化状态的系统检测子状态。

10. SSD 更新消息

    如果在接入尝试处理过程中移动台接收到这条消息，或由于关机初始化注册接收，则移动台会忽略该消息。如果在确认移动台在该子状态下发送的所有消息之后接收到该消息或者该消息中包括确认，移动台会利用 5.3.1 中描述的过程按更新共享秘密数据（SSD）说明响应消息。

11. 其他消息

    如果接收到表 5.2 中描述的消息类型的其他消息，移动台会处理消息的所有 2 层字段而忽略其他字段。而其他所有消息都被移动台忽略。

5.3.8 移动台消息发送子状态

在移动台消息发送子状态中，移动台向基站发送数据突发消息。如果基站响应带有鉴别请求，移动台在该子状态中响应，虽然该功能是可选的。

一旦进入消息发送子状态，移动台利用 5.3.1 描述的过程发送数据突发消息。

在移动台消息发送子状态中，移动台检测寻呼信道。如果移动台宣布失去寻呼信道，则发送失效并进入空闲状态。如果移动台接收在该子状态下移动台发送的所有消息的确认，移动台结束接入尝试。如果信道指配消息、鉴别查询消息、基站查询设置指令或 SSD 更新消息中不包括确认，移动台进入空闲状态。

如果接收寻呼消息或时隙寻呼消息，移动台将自身 MIN 和消息中所有记录的 MIN 进行比较。如果匹配，移动台放弃所有正在处理过程中的接入尝试并进入寻呼响应子状态，而且将消息储存以供以后发送。

如果接收到目的地址为移动台的如下消息，移动台会按如下描述的过程处理。

1. 鉴别查询消息

   如果在接入尝试处理过程中接收到鉴别查询消息，移动台忽略该消息。如果在确认移动台在该子状态下发送的所有消息之后接收到该消息或者该消息中包括确认，移动台利用 5.3.1 中描述的过程，按照惟一查询响应过程说明来响应消息。

2. 基站查询确认指令

   如果在接入尝试处理过程中接收到鉴别查询消息，移动台忽略该消息。如果在确认移动台在该子状态下发送的所有消息之后接收到该消息或者该消息中包括确认，移动台利用 5.3.1 中讨论的接入过程，按照更新共享数据（SSD）说明响应消息。

3. 数据突发消息

没有要求

4. 本地控制指令

没有要求

5. 锁指令

移动台使发送失效并在移动台半永久存储器中记录该指令的原因。移动台应该通知用户被锁条件并进入初始化状态的系统检测子状态，直到移动台再次开机或接收到解锁指令，移动台才会重新进入系统接入状态。这一要求优先于其他移动台进入系统接入状态的要求。

6. 保持要求指令

移动台在半永久存储器中记录保持必需指令的原因。移动台保持解锁状态并通知用户保持必需的状态。

7. SSD 更新消息

如果在接入尝试处理过程中接收到 SSD 更新消息，移动台会忽略该消息。如果在确认移动台在该子状态下发送的所有消息之后接收到该消息或者该消息中包括确认，移动台利用 5.3.1 中描述的接入过程，按照更新共享秘密数据说明响应消息。

8. 其他消息

如果移动台接收到表 5.2 中描述的消息类型的其他消息，会处理消息中 2 层的所有字段而忽略其他字段。移动台会忽略其他消息。

5.4　移动台控制业务信道状态

在移动台控制业务信道状态中，移动台利用前向、反向业务信道与基站通信。

如图 5.6 所示，移动台控制业务信道状态由如下几个子状态组成：

- 业务信道初始化子状态

  在业务信道初始化子状态中，移动台检验可以接收前向业务信道并开始在反向业务信道上发送。
- 等待指令子状态

  在等待指令子状态中，移动台等待信息消息告警。
- 等待移动台响应子状态

  在等待移动台响应子状态中，移动台等待用户响应呼叫。
- 会话子状态

  在会话子状态中，移动台的初步服务选项应用与基站交换初步业务包。
- 释放子状态

  在释放子状态中，移动台中断呼叫。

在描述移动台控制业务信道状态之前，先讨论前向业务信道功率控制、服务选项以及确认过程。

5.4.1　前向业务信道功率控制

为了保证前向业务信道功率控制，移动台向基站报告帧错误速率统计，可以进行周期报告或门

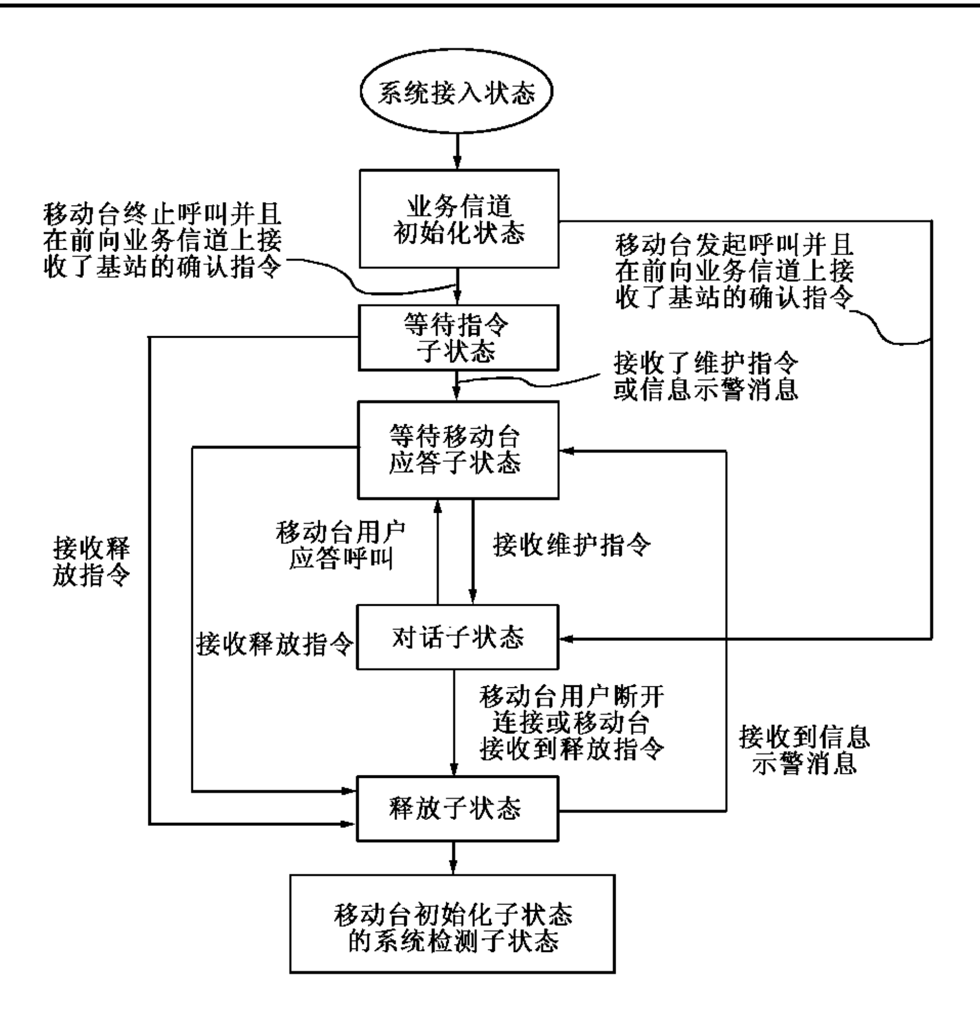


图 5.6　移动台控制业务信道状态

限报告。

移动台维持接收帧总数以及接收的坏帧数。移动台对接收到的帧采用如下处理：

- 移动台将储存的帧总数增 1。
- 如果接收的帧是坏帧，移动台将储存的坏帧数增 1。
- 如果(1)门限报告模式标识等于 1 而且储存的坏帧数等于储存的功率控制报告门限，或者(2)周期报告模式标识等于 1 而且储存的帧总数等于$\lfloor 2^{a/2} \times 5 \rfloor$，移动台就向基站发送功率测试报告消息，$a$ 表示功率控制报告帧数。
- 如果储存的帧总数等于$\lfloor 2^{a/2} \times 5 \rfloor$，移动台就将储存的帧总数和储存的坏帧数设为 0。

为了初始化前向业务信道功率控制，移动台将接收到的帧总数和接收到的坏帧数计为零并储存从功率控制消息得到的如下参数：

- 功率控制报告门限
- 功率控制报告帧数
- 门限报告模式标识
- 周期报告模式标识
- 功率报告延时

5.4.2　服务选项

在业务信道操作期间，移动台和基站支持基本业务服务。服务选项要求决定了移动台和基站

处理前向和反向业务信道帧上基本业务比特的方式。

移动台和基站都可以请求服务选项。移动台可以在响应寻呼、业务信道操作期间、呼叫开始时间内要求特定的服务选项。如果服务选项要求对基站是可以接收的，移动台和基站就开始使用新的服务选项。如果基站不能接收移动台要求的服务选项，基站可以拒绝服务选项请求或者请求一替换的服务选项。如果基站请求了替换服务选项，移动台可以接收或拒绝基站的替换服务选项，还可以请求另一服务选项。最终移动台和基站都应寻找到双方都接收的服务选项或者移动台拒绝基站的服务选项请求或基站拒绝移动台的服务选项请求。

移动台和基站利用服务选项要求指令请求服务选项或建议替换服务选项，利用服务选项响应接收或拒绝服务选项请求。

另外，移动台可以在呼叫消息或寻呼响应消息中请求服务选项，基站可以在寻呼消息或时隙寻呼消息中请求服务选项。移动台和基站利用服务消息控制指令调用服务消息特殊函数。

移动台利用变量（服务选项请求数）记录呼叫消息、寻呼响应消息或服务消息请求指令中发送的突出请求的服务选项数。在移动台没有突出服务消息请求时，服务消息请求数是 NULL。移动台利用另一变量记录当前激活的服务消息数，如果当前没有激活服务选项则该变量为 NULL。

5.4.3　处理服务选项指令

移动台接收到服务选项请求指令后，按如下过程处理：

- 如果接受所请求的服务选项，移动台将服务选项请求数设为 NULL，并在 T_{58m} = 5 秒内发送一服务选项响应指令接收请求的服务选项。T_{58m} 表示移动台对服务选项请求响应的最大时间。移动台依照请求的服务选项要求解释服务选项请求指令的消息行动时间并开始按照要求利用请求的服务选项。如果服务选项激活了，移动台将激活服务选项数设为请求的服务选项数。
- 如果移动台不接受请求的服务选项并请求一替换服务选项，移动台将服务选项请求数设为替换的服务选项数并在 T_{58m} 秒内发送服务选项请求指令请求替换的服务选项。
- 如果移动台不接受请求的服务选项但并不请求替换服务选项，移动台将服务选项请求数设为 NULL 并在 T_{58m} 秒内发送服务选项响应指令拒绝请求。移动台按接收服务选项请求指令前的方式继续处理基本业务并保持当前状态。

移动台接收到服务选项响应指令后，按如下过程处理：

- 如果指令给定的服务选项数等于服务选项请求数，移动台将请求数设为 NULL。移动台依照给定服务选项要求解释服务选项响应指令的消息行动时间并开始按照要求利用给定的服务选项。在服务选项激活时，移动台将储存的服务选项当前值设为给定的服务选项数。
- 如果指令标识了服务选项拒绝，移动台将服务选项请求数设为 NULL。移动台按接收服务选项请求指令前的方式继续处理基本业务并保持当前状态。
- 如果指令没有标识服务选项拒绝信息而且指令描述的服务选项不等于服务选项请求数，则移动台将服务选项请求数设为 NULL 并在 T_{58m} 秒内发送移动台拒绝指令[指令限定码（8 比特）如，00000001]。移动台按接收服务选项请求指令前的方式继续处理基本业务并保持当前状态。

如果当前有一不为 NULL 的激活服务选项，移动台依照激活服务选项要求解释服务选项控制

指令的消息行动时间并按照要求处理服务选项控制指令。否则，移动台在 T_{58m} 秒内发送移动台拒绝指令[指令限定码(8 比特)如，00000001]。

为了初始化服务选项请求，移动台将服务选项请求数设为给定的服务选项数。

5.4.4 确认过程

确认过程促进移动台和基站之间的有效交换。移动台利用确认序列数、消息序列数和确认要求标识字段来检测重复消息并为确认提供参考。这些消息字段都属 2 层字段，而且确认过程也属 2 层过程。其他消息字段都属 3 层字段，对 3 层的处理属 3 层处理。

在前向业务信道和反向业务信道上，处理要求确认的消息使用可选的重复策略，如果没有接收到消息的确认，则重发消息。

在确认要求标识字段设为 1 时，业务信道消息要求确认。

2 层协议不保证以任何顺序发送。如果移动台要求接着以一定顺序接收一串消息，移动台在发送下一消息时，必须等待前一消息的确认。对要求确认而相对顺序并不重要的消息，移动台可以在接收到第一个消息的确认之前最多发送 4 个这样的消息。

移动台为要求确认的消息储存消息序列数。而且为反向业务信道消息序列数字段的所有可能值储存确认状态标识。只有当确认状态标识[(消息序列确认数 +4)mod 8]为 YES 时，移动台才发送新的要求确认的消息。

当移动台在前向业务信道上接收到消息，将确认状态标识设为 NO。当移动台在反向业务信道上发送一新的要求确认的消息，将确认状态标识设为 YES 并将该消息的消息序列数字段设为消息序列确认数。接着，移动台将储存的消息序列确认数加 1 模 8。对移动台已经接收到确认的任何消息，移动台不会重发。

如果在发送消息之后的 T_{1m} = 4 秒内没有接收到确认，移动台重发该消息(如图 5.7 所示)。如果重发消息，移动台为重发使用同样的消息序列号。在前一消息发送之后的 T_{1m} 秒内，移动台不会重发该消息。T_{1m} 表示移动台等待确认的最长时间。

移动台储存所发送的要求确认消息的重发计数器。在第一次发送消息之前，移动台将重发计数器设为 0。在发送消息后如果没有接收到确认，移动台将重发计数器加 1。当重发计数器等于 N_{1m} = 3 时，移动台宣告确认失败。N_{1m} 表示移动台在反向业务信道上发送要求确认消息的最多次数。

前向业务信道上接收消息所包含的消息序列号字段按照与反向业务信道上发送消息一样的原则增加。对前向业务信道上要求确认的消息和不要求确认的消息采用不同方式的序列号。

移动台通过发送确认序列号字段等于接收消息序列号字段的消息来确认接收到的消息。以这种方式设置确认序列号字段的消息就包含了接收消息的确认。

只要接收到要求确认的消息，移动台将随后反向业务信道消息的确认序列号字段设为接收消息序列号。如果没有接收到消息，移动台将该字段设为 111。接收到要求确认的消息后，移动台在 T_{2m} = 0.2 秒内发送包含确认的消息，如图 5.8 所示。如果接收到要求确认的消息之后的 T_{2m} 秒内没有可用消息，移动台发送包含确认的移动台确认指令。移动台确认指令作为不要求确认的消息发送。T_{2m} 表示允许移动台发送确认的最长时间。

如果确认要求标识字段为 0，业务信道消息不要求确认。移动台为此要求确认的消息储存消

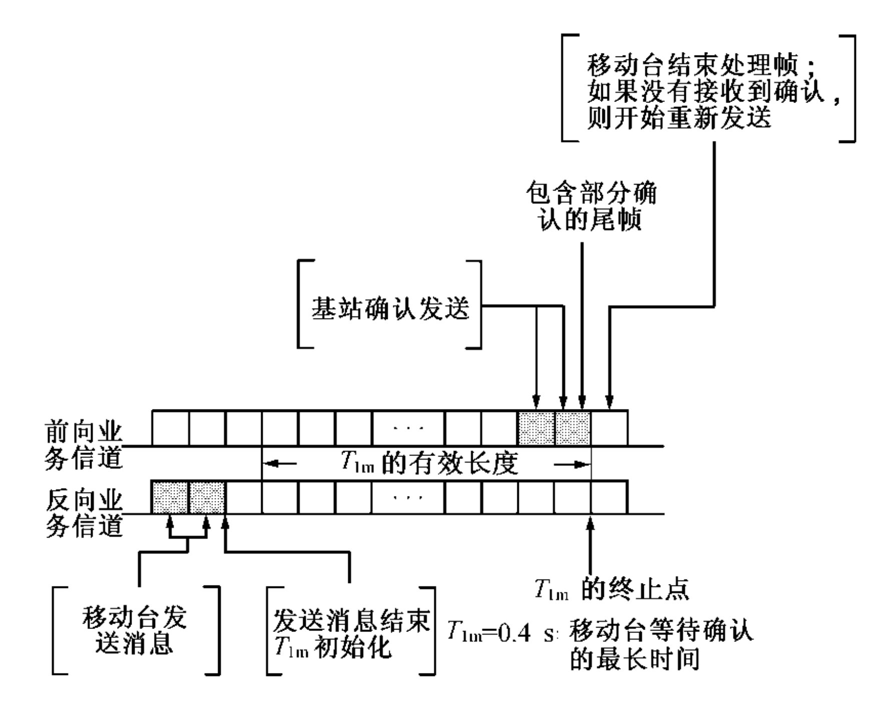


图 5.7　反向业务信道消息确认时间限制

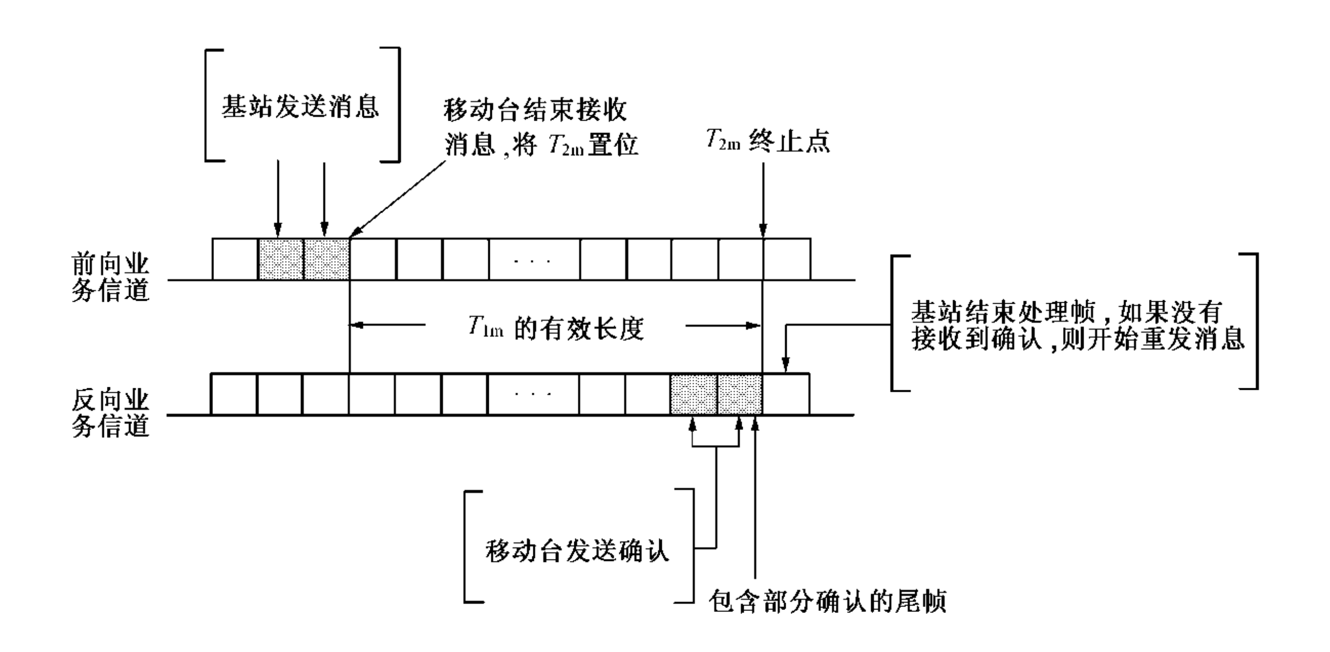
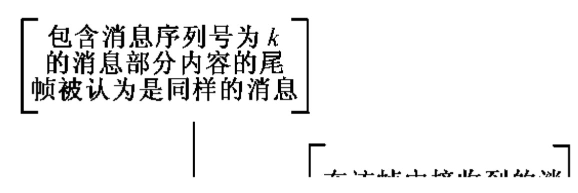


图 5.8　反向业务信道消息确认的时间限制

息序列号。对所发送的不要求确认的消息，移动台将消息的消息序列号字段设为不要求确认的号并将其（比要求确认的号）加1。

移动台认为在 T_{3m} = 0.32 秒内接收的不要求确认而且具有一样消息序列号的消息是副本，如图5.9所示。如果移动台通过消息序列号检测到其接收了一个消息的多个副本，则将副本丢弃。T_{3m}表示在前向信道上到达的两个不要求确认而且具有一样消息序列号的副本消息的周期。



移动台按如下方式重新开始确认过程：如果对所有消息序列号 n，$0 \leqslant n \leqslant 7$，的确认状态标识等于 YES，移动台保存相应消息并在完成确认过程的复位之后重发该消息。对这样的消息，将其重发计数器设为 0。移动台将下一要求确认的消息序列号设为 0，将下一不要求确认的消息序列号设为 0，将确认状态标识，$0 \leqslant n \leqslant 7$，设为 NO。

对确认序列号的复位，移动台将反向业务信道消息的确认序列号字段设为 111，直到接收到第一个要求确认的消息。

对副本检测的复位，移动台将消息序列号 n，$0 \leqslant n \leqslant 7$，的接收消息标识设为 NO。

移动台储存当前业务系统参数消息中的如下参数：

- 系统标识
- 网络标识
- 激活组、候选组、邻近组和剩余组的搜索窗大小
- 导频检测门限
- 导频下降门限
- 激活组和候选组的比较门限
- 下降定时器值
- 邻近组成员保持的最长时限

移动台终止其漫游状态并通知用户移动台是否漫游。

给定消息激活时间。对没有使用时间字段或使用时间字段为 0 的消息有内在的激活时间。而对使用时间字段为 1 的消息有在消息的激活时间字段中指定的外在激活时间。有将来激活时间的消息称为未决消息。除非另外指定，有内在激活时间的消息不迟于第一个 80 ms 边界（相对系统时间）生效，边界发生在包含消息最后一位信息的帧结束至少 80 ms 之后。有外在激活时间的消息在系统时间（以 80 ms 为单位）模 64 等于消息激活时间字段时生效。激活时间和包含消息最后一位信息的帧结束之间的时间差至少是 80 ms。移动台在不包括未决服务选项控制指令的任何给定时间支持未决消息。移动台要求支持的未决服务选项控制指令数目在服务选项中指定。

一旦接收长码转换请求指令，移动台实施如下过程。如果长码转换请求指令要求转换成私有长码而移动台能产生私有长码，移动台接受请求并在 T_{56m} = 0.2 秒内发送长码转换响应指令 00000011。其中 T_{56m} 表示对前向业务信道上接收消息或指令的默认最大响应时间。移动台在前向和反向业务信道上使用秘密长码。如果使用时间等于 0，在响应转换结束之后 T_{4m} = 20 帧的第一个 80 ms 边界（相对系统开始时间）开始使用私有长码。T_{4m} 表示响应长码转换请求指令（为秘密长码转换检测 80 ms 边界）之后的帧数。移动台应提示用户语音私有模式可用。如果不能产生私有长码或移动台不接受请求，移动台会在 T_{56m} = 0.2 秒内发送长码转换响应指令 00000010。

如果长码转换请求指令请求转换成公用长码而且接受请求，移动台在 T_{56m} 秒内发送长码转换响应指令 00000011。移动台应提示用户语音私有模式不可用。如果长码转换请求指令请求转换成公用长码而移动台不接受请求，移动台会在 T_{56m} 秒内发送长码转换响应指令 00000011。

5.4.5　业务信道初始化子状态

在业务信道初始化子状态中，移动台确认能接收前向业务信道和在反向业务信道上开始发送。

1. 一旦进入业务信道初始化子状态，移动台执行如下操作：

- 移动台进行注册初始化，将确认过程复位并初始化前向业务功率控制。
- 移动台将激活选项号设为 NULL，表示当前没有激活服务选项。
- 如果呼叫由移动台发起而且发起消息请求一特定服务选项，移动台执行服务选项请求初始化，指定特定服务选项号。
- 如果呼叫由移动台发起而发起消息没有请求特定服务选项，移动台执行服务选项请求初始化，指定 1（默认服务选项号）。
- 如果呼叫被移动台终止，移动台执行服务选项请求初始化，指定寻呼响应消息中请求的服务选项号。

2. 在业务信道初始化子状态中，移动台执行如下操作：

- 移动台执行导频能力测试但并不发送导频能力测试消息。
- 移动台执行注册计数器保持。

  如果不支持指配的 CDMA 信道或指配的前向业务编码信道，移动台进入初始化状态的系统检测子状态。

3. 如果支持指配的 CDMA 信道和指配前向业务编码信道，移动台执行如下操作：

- 移动台调谐到指配的 CDMA 信道。
- 移动台将其编码信道设为指配的前向业务编码信道。
- 移动台将前向和反向业务信道偏移设为当前业务信道帧偏移确定的指配帧偏移，以 1.25 微秒为单位。
- 移动台将其前向和反向业务信道长码掩码设为公用长码掩码。

  如果移动台在进入子状态之后的 T_{50m} = 0.2 秒内没有接收到 N_{5m} = 2 个连续好帧，移动台进入初始化状态的系统检测子状态。N_{5m} 表示进入移动台控制业务信道状态的业务信道初始化子状态后，允许移动台启用发送器之前接收的连续好的前向业务信道帧数。T_{50m} 表示在业务信道初始化子状态中获得 N_{5m} 个连续好的前向业务信道帧的最长时间。

4. 如果在进入子状态的 T_{50m} 秒内接收到 N_{5m} 个连续好帧，移动台在保持业务信道初始化子状态情况下执行如下附加过程：

- 移动台执行前向业务信道管理。如果声明失去前向业务信道，移动台进入初始化状态的系统检测子状态。
- 移动台调整发送功率。
- 移动台发送业务信道序。
- 移动台执行确认过程。如果宣告确认失败，移动台使发送器失效并进入初始化状态的系统检测子状态。

如果移动台在进入该子状态的 T_{51m} = 2 秒内没有接收到基站确认指令，移动台使发送器失效并进入移动台初始化状态的系统检测子状态。T_{51m} 表示在业务信道初始化子状态中，移动台接收基站确认指令的最长时间。

5. 如果移动台在进入该子状态的 T_{51m} 秒内接收到基站确认指令，移动台执行如下过程：

- 如果呼叫由移动台终止，移动台进入等待指令子状态

- 如果呼叫由移动台发起，移动台进入会话子状态。

5.4.6　等待指令子状态

在等待指令子状态中，移动台等待信息消息告警。一旦进入等待指令子状态，移动台将子状态计数器设为 T_{52m} = 5 秒。T_{52m}表示在移动台控制业务信道状态的等待指令子状态中（将移动台转变到不同子状态或状态）接收消息的最长时间。

1. 在等待指令子状态中，移动台执行如下过程：

- 如果子状态计数器满，移动台使发送器失效并进入初始化状态的系统检测子状态。
- 移动台执行前向业务信道管理。如果声明失去前向业务信道，移动台进入初始化状态的系统检测子状态。
- 移动台调整发送功率，执行前向业务信道功率控制和传递过程。
- 如果有激活服务选项（激活服务选项号不等于 NULL），移动台按照激活服务选项的要求处理接收的基本业务比特；否则，移动台丢弃接收的基本业务比特。
- 如果有激活服务选项（即激活服务选项号不等于 NULL），移动台按照激活服务选项的要求发送基本业务比特；否则，移动台发送空的业务信道数据。
- 移动台执行注册计数器保持。
- 如果用户要求移动台发送消息，移动台发送数据突发消息。
- 如果用户要求移动台请求服务选项，移动台执行服务选项请求初始化指定请求的服务选项号并发送服务选项请求指令。
- 如果有激活服务选项（即激活服务选项号不等于 NULL），移动台发送服务选项控制指令按照激活服务选项要求调用服务选项特定的函数。
- 如果用户要求移动台请求私有长码转换而且移动台有长码掩码，移动台发送要求确认的长码转换请求指令 00000001。
- 如果用户要求移动台请求公用长码转换，移动台发送要求确认的长码转换请求指令 00000000。
- 如果用户要求移动台工作于模拟模式，移动台发送要求确认的请求模拟服务指令。
- 如果用户要求移动台关机，移动台带着关机指示进入释放子状态。
- 移动台执行确认过程。如果宣告确认失败，移动台使发送器失效并进入初始化状态的系统检测子状态。
- 如果移动台接收如下列表中的消息而且消息字段值都在允许范围内，移动台按照如下描述过程和消息激活时间处理消息。

  1. 信息消息告警

     如果消息包含信号信息记录，移动台应按照信号信息记录对用户告警；否则，移动台利用标准告警。移动台应进入等待移动台答复子状态。

  2. 模拟切换指示消息

     基站指示移动台通过发送模拟切换指示消息来执行 CDMA 模拟切换。移动台按照先前描述方式处理消息并进入等待带有从 CDMA 标识切换的指令掩码的状态。

  3. 审查指令
  4. 鉴别查询消息

移动台将子状态计数器设为 T_{52m} = 5 秒。移动台处理消息然后在 T_{32m} = 5 秒内响应消息。T_{32m} 表示对 SSD 更新消息、基站查询设置指令和鉴别查询消息响应的最长时间。

5. 基站确认指令
6. 基站确认查询指令

   移动台将子状态计数器设为 T_{52m} 秒。接着移动台处理消息并在 T_{32m} 秒内通过 SSD 更新确认指令或 SSD 更新拒绝指令回复消息。

7. 数据突发消息
8. 切换指示消息

   移动台处理消息并将子状态计数器设为 T_{52m} 秒。

9. 业务系统参数消息

   移动台按照"处理业务系统参数消息"已经描述的方式处理消息。

10. 本地控制指令
11. 锁指令

    移动台使发送器失效并在移动台半永久存储器中记录该指令的原因。移动台应该通知用户被锁状态并进入初始化状态的系统检测子状态。直到移动台再次开机或接收到解锁指令，移动台才会重新进入系统接入状态。这一要求优先于其他移动台进入系统接入状态的要求。

12. 长码转换请求指令

    如果长码转换请求指令请求转换成私有长码而且移动台可以产生私有长码，移动台接受请求并在 T_{56m} = 0.2 秒内发送长码转换响应指令 00000011。

13. 保持指令

    移动台进入等待移动台答复子状态。

14. 保持要求指令

    移动台在半永久存储器中记录保持必需指令的原因。移动台保持解锁条件并通知用户保持必需的状态。

15. 消息加密模式指令

    增强确认过程和保护敏感预定信息的一种方法是对所选业务信道信号消息的一些特定字段进行加密。每次呼叫单独控制信号消息加密。呼叫的初始加密模式由信道指配消息中的加密模式字段值决定。如果加密模式设为 0000，消息加密无效。信道指配后基站通过发送加密模式消息或将加密模式字段设为 0001 将加密设为有效。

16. 移动台注册消息
17. 邻近列表更新消息

    移动台支持大小至少为 N_{8m} = 20 个导频的邻近组，N_{8m} 表示所支持的邻近组的最大数。在移动台首次指配前向业务信道时初始化邻近组，让其包括最近接收的邻近列表消息中描述的导频。移动台为邻近组中所有导频保持计数器（邻近列表时限）。在移动台将导频从激活组或候选组移到邻近组时，将计数器初始化为 0。一旦接收到邻近列表更新消息，移动台将邻近组中导频的计数器增 1 并从邻近组中

删除所有计数器超过邻近组最长时限的导频。

18. 参数更新指令

移动台将子状态计数器重新设为 T_{52m} 秒。移动台增加一个模 64 计算。移动台在 T_{56m} 秒内发送参数更新设置指令。移动台将参数更新设置指令中的指令限制编码字段设置成与参数更新指令中的指令限制编码字段值相等。

19. 导频测量请求指令

如果接收到导频测量请求指令，移动台在 T_{56m} 秒内发送导频强度测量消息。

20. 功率控制参数消息

移动台储存如下功率控制参数消息中的参数（储存值 = 接收值）

- 功率控制报告门限
- 功率控制报告数
- 门限报告模式标识
- 周期报告模式标识
- 功率报告时延

移动台设置接收的帧总数以及接收的坏帧总数。

21. 释放指令

移动台带着基站释放标识进入释放子状态。

22. 取参数消息

移动台在 T_{56m} 秒内发送参数响应消息。

23. 服务选项控制指令

如果当前有一激活服务选项，移动台依照激活服务选项要求解释服务选项控制指令的消息行动时间并按照要求处理服务选项控制指令。否则，移动台在 T_{58m} 秒内发送移动台拒绝指令 00000001。

24. 服务选项请求指令

如果接受所请求的服务选项，移动台将服务选项请求数设为 NULL 并在 T_{58m} 秒内发送一服务选项响应指令接受请求的服务选项。如果移动台不接受请求的服务选项并请求一替换服务选项，移动台将服务选项请求数设为替换的服务选项数并在 T_{58m} 秒内发送服务选项请求指令请求替换的服务选项。如果移动台不接受请求的服务选项但并不请求替换服务选项，移动台将服务选项请求数设为 NULL 并在 T_{55m} = 2 秒内发送服务选项响应指令拒绝请求。

25. 服务选项响应指令

如果指令给定的服务选项数等于服务选项请求数，移动台将请求数设为 NULL。如果指令标识了服务拒绝，移动台将服务选项请求数设为 NULL。如果指令没有标识服务选项拒绝信息而且指令描述的服务选项不等于服务选项请求数，则移动台将服务选项请求数设为 NULL 并在 T_{58m} = 5 秒内发送移动台拒绝指令 00000100。

26. 设置参数消息

如果移动台能设置消息中参数 ID 字段描述的所有参数，则设置；否则，移动台在 T_{56m} 秒内发送移动台拒绝指令。

27. SSD 更新消息

移动台将子状态计数器重新设为 T_{52m} 秒。接着移动台处理消息并在 T_{32m} 秒内通过基站查询指令响应。

28. 状态请求指令

移动台在 T_{56m} 秒内发送状态消息。

- 如果移动台接收了其他消息，处理消息的所有 2 层字段。如果接收了不能处理的消息，移动台丢弃消息并在 T_{56m} 秒内发送移动台拒绝指令。

5.4.7 等待移动台答复子状态

在等待移动台答复子状态中，移动台等待用户响应移动台终止呼叫。

一旦进入等待移动台答复子状态，移动台将子状态计数器设为 T_{53m} = 65 秒。T_{53m} 表示在等待移动台答复子状态中接收使移动台转换子状态的消息的最长时间。在等待移动台答复子状态中，移动台执行如下操作：

- 如果子状态计数器满，移动台使发送器失效并进入初始化状态的系统检测子状态。
- 移动台执行前向业务信道管理。如果宣布失去前向业务信道，移动台进入初始化状态的系统检测子状态。
- 移动台调整发送功率。工作在 CDMA 发送模式，移动台提供两个独立的调整输出功率的方法，即开环预测（仅仅是移动台操作）和闭环修正（包含移动台和基站）。
- 在移动台控制前向业务信道状态时，移动台执行切换过程。

  1. 软切换：在软切换中，移动台并不中断与原基站通信而开始与新基站通信。软切换仅用于同一载频 CDMA 信道。
  2. CDMA 到 CDMA 的硬切换：不同基站不同载频或不同帧偏移中跃迁的切换。
  3. CDMA 到模拟的切换：在此切换中，移动台从前向业务信道中转向模拟语音信道。

- 如果有激活服务选项（激活服务选项号不等于 NULL），移动台按照激活服务选项的要求处理接收的基本业务比特；否则，移动台丢弃接收的基本业务比特。
- 如果有激活服务选项（即激活服务选项号不等于 NULL），移动台按照激活服务选项的要求发送基本业务比特；否则，移动台发送空的业务信道数据。
- 移动台执行注册计数器保持。移动台提供使计数器有效和失效的方法。当一计数器失效，则不会认为该计数器满。有效的计数器就是激活计数器。移动台提供如下注册计数器：

  1. 开机/初始化计数器
  2. 基于计数器注册的计数器
  3. 区域列表记录计数器
  4. SID/NID 列表记录计数器

- 如果用户指示移动台响应呼叫，移动台向基站发送要求消息确认的连接指令。移动台进入会话子状态。
- 如果用户指示移动台发送消息，移动台发送数据突发消息。
- 如果用户指示移动台请求服务选项，移动台执行服务选项请求初始化指定的所请求的服务

选项数并发送服务选项请求指令。

- 如果有激活服务选项，移动台可能发送服务选项控制指令按照激活服务选项要求调用服务选项特定函数。
- 如果用户指示移动台请求私有长码转换而且移动台有长码掩码，移动台发送要求确认的长码转换请求指令00000001。
- 如果用户指示移动台请求公用长码转换，移动台发送要求确认的公用长码转换请求指令00000000。
- 如果用户指示移动台操作于模拟模式，移动台发送要求确认的请求模拟服务的指令。
- 如果用户指示移动台关机，移动台进入带有关机标识的释放子状态。
- 移动台按5.3.2中的描述处理确认过程。如果宣布确认失败，移动台使发送器失效并进入初始化状态的系统检测子状态。
- 如果移动台接收了如下列表中包括的消息，而且消息的所有字段都在允许范围内，移动台按照下面描述过程和消息激活数据处理消息。

1. 信息消息告警

   移动台将子状态计数器设为T_{55m}秒。如果信息警告消息不含信号信息记录，移动台使用标准警告。

2. 模拟切换指示消息

   基站通过发送模拟切换指示消息指示移动台执行从CDMA到模拟的切换。在模拟切换指示消息指定的反应时间里，移动台使发送器失效。在$T_{63m}=0.1$秒内移动台将发送器在模拟语音信道上确立。T_{63m}表示执行CDMA到模拟切换的最长时间。移动台处理完消息进入等待答复任务。

3. 审查指令
4. 确认查询消息

   基站产生24位随机查询值并在寻呼信道或前向业务信道中的确认查询消息中发送给移动台。一旦接收到确认查询消息，移动台初始化确认算法。移动台处理该消息并在T_{32m}秒内响应。

5. 基站确认指令
6. 基站确认查询指令

   移动台处理消息并在T_{32m}秒内做出SSD更新确认指令或在T_{32m}秒内做出SSD更新拒绝指令的响应。

7. 数据突发消息
8. 切换指示消息

   如果接收到切换指示消息，移动台按如下描述处理。当消息见效，移动台按照切换指示消息更新激活组，而且不再继续使用导频不在切换指示消息之列的前向业务信道。如果接收到的当前业务信道帧偏移不等于储存的当前业务信道帧偏移，移动台改变前向和反向业务信道的帧偏移。

9. 业务系统参数消息

   移动台储存系统参数消息中的如下参数：

- 系统和网络标识
- 激活组、候选组、连接组以及剩余组的搜索窗大小
- 导频检测门限
- 导频下降门限
- 相对候选组的激活组

10. 本地控制指令
11. 锁指令

移动台使发送失效并在移动台半永久存储器中记录该指令的原因。移动台应该通知用户被锁状态并进入初始化状态的系统检测子状态。直到移动台再次开机或接收到解锁指令,移动台才会重新进入系统接入状态。这一要求优先于其他移动台进入系统接入状态的要求。

12. 长码转换请求指令

如果长码转换请求指令要求转换成私有长码而移动台能产生私有长码,移动台接受请求并在 T_{56m} 秒内发送长码转换响应指令 00000011。如果长码转换请求指令请求转换成公用长码而且接受请求,移动台在 T_{56m} 秒内发送长码转换响应指令 00000010。

13. 保持指令

移动台将子状态计数器设为 T_{56m} 秒。

14. 保持要求指令

移动台在半永久存储器中记录保持必需的指令原因。移动台保持解锁条件并将保持要求状态通知用户。

15. 消息加密模式指令

提供确认过程和保护敏感预定信息的一种方法是对所选业务信道信号消息的一些特定字段进行加密。每次呼叫单独控制信号消息加密。呼叫的初始加密模式由信道指配消息中的加密模式字段值决定。如果加密模式设为 0000,消息加密无效。信道指配后基站通过发送加密模式字段为 0001 的切换指示消息将加密设为有效。

基站也发送加密模式字段为 0001 的消息加密模式指令。

16. 移动台注册消息

当基站确认移动台注册,移动台在前向业务信道上接收含有位置和其他参数的移动台注册消息。

17. 邻近列表更新消息

移动台支持大小至少为 N_{8m} = 20 个导频的邻近组,N_{8m} 表示所支持的邻近组的最小数。移动台在首次指配前向业务信道时初始化邻近组,让其包括最近接收的邻近列表消息中描述的导频。移动台为邻近组中所有导频保持计数器。在移动台将导频从激活组或候选组移到邻近组时,将计数器初始化为 0。一旦接收到邻近列表更新消息,移动台将邻近组中导频的计数器增 1 并从邻近组中删除所有计数器超过邻近组最长时限的导频。

18. 参数更新指令

移动台在半永久储存器中增加呼叫参数。移动台在 T_{56m} = 0.2 秒内发送参数更新确认指令。移动台让参数更新确认指令中的字段值等于参数更新指令中的字段值。

19. 导频测量请求指令

如果接收到导频测量请求指令，移动台在 T_{56m} 秒内发送导频电平测量消息。

20. 功率控制参数消息

移动台储存如下功率控制参数消息：

- 功率控制报告门限
- 功率控制报告帧数
- 门限报告模式标识
- 周期报告模式标识
- 功率报告时延

移动台设置接收的帧总数以及接收的坏帧总数。

21. 释放指令

移动台通过基站释放标识进入释放子状态。

22. 响应参数消息

移动台在 T_{56m} 秒内发送参数响应消息。

23. 服务选项控制指令

如果当前有一激活服务选项，移动台依照激活服务选项要求解释服务选项控制指令的消息行动时间并按照要求处理服务选项控制指令。否则，移动台在 T_{56m} 秒内发送移动台拒绝指令 00000001。

24. 服务选项请求指令

当移动台接收到服务选项请求指令，执行如下操作：

- 如果接受所请求的服务选项，移动台将服务选项请求数设为 NULL 并在 T_{56m} = 5 秒内发送一服务选项响应指令接受请求的服务选项。移动台依照请求的服务选项要求解释服务选项请求指令的消息行动时间并开始按照要求利用请求的服务选项。如果服务选项激活了，移动台将激活服务选项数设为请求的服务选项数。
- 如果移动台不接受请求的服务选项并请求一替换服务选项，移动台将服务选项请求数设为替换的服务选项数并在 T_{58m} 秒内发送服务选项请求指令请求替换服务选项。
- 如果移动台不接受请求的服务选项且并不请求替换服务选项，移动台将服务选项请求数设为 NULL 并在 T_{58m} 秒内发送服务选项响应指令拒绝请求。移动台按接收服务选项请求指令前的方式继续处理基本业务并保持当前状态。

25. 服务选项响应指令

移动台接收到服务选项响应指令后，按如下过程处理：

- 如果指令给定的服务选项数等于服务选项请求数，移动台将请求数设为 NULL。移动台依照给定服务选项要求解释服务选项响应指令的消息行动时间并开始按照要求利用给定的服务选项。在服务选项激活时，移动台令激活服务选项数等于特定的服务选项数。
- 如果指令标识了服务拒绝，移动台将服务选项请求数设为 NULL。移动台按接收服务选项请求指令前的方式继续处理基本业务并保持当前状态。

- 如果指令没有标识服务选项拒绝信息而且指令描述的服务选项不等于储存的服务选项请求数，则移动台将服务选项请求数设为 NULL 并在 T_{58m} 秒内发送移动台拒绝指令 00000100。移动台按接收服务选项请求指令前的方式继续处理基本业务并保持当前状态。

26. 设置参数消息

如果移动台能设置消息中参数 ID 字段描述的所有参数，则设置；否则，移动台在 T_{56m} 秒内发送移动台拒绝指令。

27. SSD 更新消息

移动台处理消息并在 T_{32m} 秒内通过基站查询指令响应。

28. 状态请求指令

移动台在 T_{56m} 秒内发送状态消息。

- 如果移动台接收了其他属于表 5.2 中描述类型的消息，处理消息的所有 2 层字段。如果接收的消息不在以上列表中或无法处理，移动台丢弃消息并在 T_{56m} 秒内发送移动台拒绝指令。

5.4.8　会话子状态

在会话子状态中，移动台基础业务服务选项应用与基站交换基础业务比特。在会话子状态中，移动台执行如下操作：

- 移动台执行前向业务信道管理。如果声明失去前向业务信道，移动台进入初始化状态的系统检测子状态。
- 移动台调整发送功率并按 5.4.1 的描述执行前向业务信道功率控制。
- 移动台按 5.2.4 的描述执行切换过程。
- 如果激活服务选项号不等于 NULL，移动台按照激活服务选项的要求处理接收的基本业务比特；否则，移动台丢弃接收的基本业务比特。
- 如果有激活服务选项（即激活服务选项号不等于 NULL），移动台按照激活服务选项的要求发送基本业务比特；否则，移动台发送空的业务信道数据。
- 移动台执行注册计数器保持。
- 如果移动台发起呼叫，且在发起消息中没有发送所有的拨号数字，则移动台在发起继续消息中发送剩下的拨号数字。在进入会话子状态的 $T_{54m} = 0.2$ 秒内，移动台发送要求确认的发起继续消息。
- 如果用户要求移动台发送消息，移动台发送数据突发消息。
- 如果用户要求移动台请求服务选项，移动台执行服务选项请求初始化指定请求的服务选项号并发送服务选项请求指令。
- 如果激活服务选项号不等于 NULL，移动台发送服务选项控制指令按照激活服务选项要求调用服务选项特定的函数。
- 如果用户要求移动台请求私有长码转换而且移动台有长码掩码，移动台发送要求确认的长码转换请求指令 00000001。
- 如果用户要求移动台请求公用长码转换，移动台发送要求确认的长码转换请求指令

00000000。

- 如果用户要求移动台发出闪光，移动台建立有信息消息的闪光，而且收集的数字包含在工具信息记录中，移动台将该消息发送给基站并要求确认。
- 如果用户要求移动台发送突发双音多频（DTMF）数字，移动台建立拨号数字的突发 DTMF 发送消息，将该消息发送给基站并要求确认。
- 如果用户要求移动台发送连续双音多频（DTMF）数字，移动台建立拨号数字的 DTMF 语音指令，将该消息发送给基站并要求确认。当用户要求移动台停止发送连续 DTMF 数字，移动台将发送连续 DTMF 语音指令 11111111 并要求确认。
- 如果用户要求移动台工作于模拟模式，移动台发送要求确认的请求模拟服务指令。
- 如果用户要求移动台断开呼叫，移动台带着移动台释放指示进入释放子状态。
- 如果用户要求移动台关机，移动台带着关机指示进入释放子状态。
- 移动台执行确认过程。如果宣告确认失败，移动台使发送器失效并进入初始化状态的系统检测子状态。
- 如果移动台接收如下列表中的消息而且消息字段值都在允许范围内，移动台按照如下描述过程和消息激活时间处理消息。

1. 信息消息告警

   如果消息包含信号信息记录而信号类型字段设为 01 或 10 或者消息不包含信号信息记录，移动台进入等待移动台答复子状态。如果信息消息警告不包含信号信息记录，移动台使用标准告警。
2. 模拟切换指示消息

   移动台处理消息并通过从 CDMA 切换的标识进入会话任务。
3. 审查指令
4. 鉴别查询消息

   移动台处理消息并在 T_{32m} 秒内响应消息。基站产生 24 位随机查询值并在寻呼信道或前向业务信道中的确认查询消息中发送给移动台。一旦接收到确认查询消息，移动台初始化确认算法。
5. 基站确认指令
6. 基站查询确认指令

   移动台处理消息并在 T_{32m} 秒内做出 SSD 更新确认指令或 SSD 更新拒绝指令的响应。
7. 连续 DTMF 语音指令

   该功能是可选的。
8. 数据突发消息
9. 信息消息闪光
10. 切换指示消息

    如果移动台接收到切换指示消息，移动台按如下描述过程处理消息。

    - 如果接收的业务信道帧偏移不等于储存的业务信道帧偏移而且使用激活时间标识（1 比特）等于 0，移动台在第一个 80 ms 边缘（相对系统时间）处理消息，这发生在包含消息最后比特的帧结束至少 80 ms 后。

- 当消息见效，移动台按照切换指示消息更新激活组，而且不再继续使用未列在切换指示消息之列的前向业务信道导频。
- 如果接收到的当前业务信道帧偏移不等于储存的当前业务信道帧偏移，移动台改变前向和反向业务信道的帧偏移。
- 如果反向 2 层标识（1 比特）等于 1，将确认过程和前向业务信道功率控制数复位。如果该字段为 1，移动台重新确定它的 2 层序列直到执行切换。

11. 业务系统参数消息

移动台储存当前业务系统参数消息中的如下参数：

- 系统和网络标识
- 激活组、候选组、邻近组和剩余组的搜索窗大小
- 导频检测门限
- 导频下降门限
- 激活组和候选组的比较门限
- 下降定时器值
- 邻近组成员保存的最长时限

12. 本地控制指令
13. 锁指令

移动台使发送器失效并在移动台半永久存储器中记录该指令的原因。移动台应该通知用户被锁状态并进入初始化状态的系统检测子状态。直到移动台再次开机或接收到解锁指令，移动台才会重新进入系统接入状态。这一要求优先于其他移动台进入系统接入状态的要求。

14. 长码转换请求指令

- 如果长码转换请求指令请求转换成公用长码而且接受请求，移动台在 T_{56m} 秒内发送长码转换响应指令 00000010。移动台应提示用户语音私有模式不可用。如果长码转换请求指令请求转换成公用长码而移动台不接受请求，移动台会在 T_{56m} 秒内发送长码转换响应指令 00000011。
- 如果长码转换请求指令要求转换成私有长码而移动台能产生私有长码，移动台接受请求并在 T_{56m} 秒内发送长码转换响应指令 00000011。移动台在前向和反向业务信道上使用私有长码。如果使用时间标识等于 0，在响应转换结束之后 N_{4m} 帧的第一个 80 ms 边界（相对系统开始时间）开始使用私有长码。移动台应提示用户语音私有模式可用。
- 如果长码转换请求指令要求转换成私有长码而移动台不能产生私有长码或移动台不接受请求，移动台会在 T_{56m} 秒内发送长码转换响应指令 00000010。

15. 保持指令

移动台进入等待移动台响应子状态。

16. 保持要求指令

移动台在半永久存储器中记录保持必需的指令原因。移动台保持解锁条件并通知用

户保持必需的状态。

17. 消息加密模式指令

提供确认过程和保护敏感预定信息的一种方法是对所选业务信道信号消息的一些特定字段进行加密。如果不执行确认，消息不加密。每次呼叫单独控制信号消息加密。呼叫的初始加密模式由信道指配消息中的加密模式字段（4 比特）值决定。如果加密模式设为 0000，消息加密无效，在前向和反向业务信道上发送的消息都不加密。信道指配后基站通过发送如下消息给移动台让加密有效。

- 消息加密模式字段等于 0001 的切换指示消息。
- 消息加密模式字段等于 0001 的消息加密模式指令。
- 消息加密模式字段（1 比特）设为 1 的模拟切换指示消息。如果设为 0，模拟语音信道的消息加密模式无效，如果设为 1，消息加密模式有效。

18. 移动台注册消息

当基站确认移动台注册，移动台在前向业务信道上接收含有比特值和其他参数的移动台注册消息。

19. 邻近列表更新消息

移动台支持大小至少为 N_{8m} = 20 个导频的邻近组。在移动台首次指配前向业务信道时初始化邻近组，让其包括最近接收的邻近列表消息中描述的导频。

如果移动台接收邻近列表更新消息，执行如下操作：

- 将邻近组中所有导频的邻近列表时限增 1。
- 从邻近组中删除所有计数器超过邻近组最长时限的导频。
- 向邻近组中添加消息中提到的且属于候选组和剩余组中的导频。如果移动台在邻近组中只能储存 k 个导频但在邻近列表更新消息中有多于 k 个导频，移动台就储存消息中前 k 个导频。

20. 参数更新指令

移动台增加储存在半永久储存器中的呼叫历史参数（64 比特）。移动台在 T_{56m} 秒内发送参数更新确认指令。移动台将参数更新确认指令中的指令限制编码字段设置成与参数更新指令中的指令限制编码字段值相等。

21. 导频测量请求指令

移动台在 T_{56m} 秒内发送导频电平测量消息。

22. 功率控制参数消息

移动台储存如下功率控制参数消息中的参数：

- 功率控制报告门限
- 功率控制报告帧数
- 门限报告模式标识
- 周期报告模式标识
- 报告功率时延

移动台设置接收的帧总数以及接收的坏帧总数为零。

23. 释放指令

移动台使用基站释放标识进入释放子状态。

24. 响应参数消息

移动台在 T_{58m} 秒内发送参数响应消息。

25. 发送突发 DTMF 消息

这一功能是可选的。

26. 服务选项控制指令

如果当前有一激活服务选项,移动台依照激活服务选项要求解释服务选项控制指令的消息行动时间并按照要求处理服务选项控制指令。否则,移动台在 T_{58m} 秒内发送移动台拒绝指令 00000001。

27. 服务选项请求指令

当移动台接收到服务选项请求指令,执行如下操作:

- 如果接受所请求的服务选项,移动台将服务选项请求数设为 NULL 并在 T_{58m} 秒内发送一服务选项响应指令接受请求的服务选项。移动台依照请求的服务选项要求解释服务选项请求指令的消息行动时间并开始按照要求利用请求的服务选项。如果服务选项激活了,移动台将激活服务选项数设为请求的服务选项数。
    - 如果移动台不接受请求的服务选项并请求一替换服务选项,移动台将服务选项请求数设为替换的服务选项数并在 T_{58m} 秒内发送服务选项请求指令请求替换服务选项。
- 如果移动台不接受请求的服务选项但并不请求替换服务选项,移动台将服务选项请求数设为 NULL,并在 T_{58m} 秒内发送服务选项响应指令拒绝请求。移动台按接收服务选项请求指令前的方式继续处理基本业务并保持当前状态。

28. 服务选项响应指令

移动台接收到服务选项响应指令后,按如下过程处理:

- 如果指令给定的服务选项数等于服务选项请求数,移动台将请求数设为 NULL。移动台依照给定服务选项要求解释服务选项响应指令的消息行动时间并开始按照要求利用给定的服务选项。在服务选项激活时,移动台令激活服务选项数等于特定的服务选项数。
- 如果指令标识了服务选项拒绝,移动台将服务选项请求数设为 NULL。移动台按接收服务选项请求指令前的方式继续处理基本业务并保持当前状态。
- 如果指令没有标识服务选项拒绝信息而且指令描述的服务选项不等于储存的服务选项请求数,则移动台将服务选项请求数设为 NULL 并在 T_{58m} 秒内发送移动台拒绝指令 00000100。移动台按接收服务选项请求指令前的方式继续处理基本业务并保持当前状态。

29. 设置参数消息

如果移动台能设置消息中参数 ID 字段描述的所有参数,则设置;否则,移动台在 T_{56m} 秒内发送移动台拒绝指令。

30. SSD 更新消息

移动台处理消息并在 T_{32m} 秒内使用基站查询指令响应。

31. 状态拒绝指令

移动台在 T_{56m} 秒内发送状态消息。

如果移动台接收了其他消息，如指令消息，移动台会处理消息的所有 2 层字段。如果接收的消息不在前向业务信道消息之列，移动台丢弃消息并在 T_{56m} 秒内发送移动台拒绝指令。

5.4.9　释放子状态

在释放子状态中，移动台确认呼叫断开。一旦进入释放子状态，移动台执行如下操作：

- 移动台将子状态计数器设为 T_{55m} 秒。
- 如果移动台是通过关机标识进入释放子状态，移动台发送释放指令 00000001 并执行关机注册过程。
- 如果移动台是通过释放标识进入释放子状态，移动台发送释放指令 00000000。
- 如果移动台是带着基站标识进入释放子状态，移动台发送释放指令 00000000。移动台使发送器失效并进入初始化状态的系统检测子状态。

在释放子状态中，移动台执行如下操作：

- 如果 T_{55m} 秒的计数器满，移动台使发送器失效并进入初始化状态的系统检测子状态。
- 移动台执行前向业务信道管理。如果移动台宣布失去前向业务信道，则进入初始化状态的系统检测子状态。
- 移动台调整发送功率并执行前向业务信道功率控制。
- 移动台执行切换过程。
- 移动台在反向业务信道上发送空业务信道数据。
- 移动台执行注册计数器保持。
- 移动台执行确认过程。如果宣布确认过程失败，则发送失效并进入初始化状态的系统检测子状态。
- 如果移动台接收了如下列表中包括的消息，而且消息的所有字段都在允许范围内，移动台按照下面描述过程和消息激活时间处理消息。

1. 信息消息告警

   移动台进入等待响应子状态。如果信息消息警告不包含任何信息记录，移动台使用标准告警，信号类型为 01，告警信号音调为 00 以及信号编码为 000001。信息记录允许网络将信息通过语音和其他告警信号转换给用户。

2. 基站确认指令
3. 数据突发消息
4. 切换指示消息

   如果移动台接收到切换指示消息，移动台按如下描述过程处理消息。

   - 如果接收的业务信道帧偏移不等于储存的业务信道帧偏移而且使用激活时间标识（1 比特）等于 0，移动台在第一个 80 ms 边缘（相对系统时间）处理消息，这发生在包含消息最后一位的帧结束至少 80 ms 后。

- 否则，消息在激活时间生效。

当消息见效，移动台执行如下操作：

- 按照切换指示消息更新激活组。
- 不再继续使用导频不在切换指示消息之列的前向业务信道。
- 如果接收到的当前业务信道帧偏移不等于储存的当前业务信道帧偏移，移动台改变前向和反向业务信道的帧偏移。
- 复位确认过程指令用来在移动台中复位确认过程。为了指示移动台复位确认过程，基站将该字段设为1。否则设为0。如果该字段为1，移动台将前向业务信道功率控制计数器复位。
- 处理加密模式字段。增强确认过程和包含敏感预定信息（如 PINs）的方法之一是对所选业务信道信号消息的特定字段进行加密。

5. 业务系统参数消息

移动台储存当前业务系统参数消息中的如下参数：

- 系统标识
- 网络标识
- 激活组、候选组、邻近组和剩余组的搜索窗大小
- 导频检测门限
- 导频下降门限
- 激活组和候选组的比较门限
- 下降定时器值
- 邻近组成员保持的最长时限

移动台应通知用户是否漫游。

6. 本地控制指令
7. 锁指令

移动台使发送失效并在移动台半永久存储器中记录该指令的原因。移动台应该通知用户被锁状态并进入初始化状态的系统检测子状态。直到移动台再次开机或接收到解锁指令，移动台才会重新进入系统接入状态。这一要求优先于其他移动台进入系统接入状态的要求。

8. 保持要求指令

移动台在半永久存储器中记录保持必需的指令原因。移动台保持解锁条件并将保持要求条件通知用户。

9. 移动台注册消息

当基站确认移动台注册，移动台在前向业务信道上接收含有位置和其他参数的移动台注册消息：

- 系统和网络标识
- 注册区域
- 保留注册区域数

- 区域计数器长度
- 多个 SID 储存标识
- 多个 NID 储存标识
- 基站纬度
- 基站经度
- 注册距离

移动台执行如下操作：

- 将第一个空闲 ID 状态设为可用。
- 增加基站的注册区域数，如果 SID，NID 不在列表，则增加到列表中。
- 将区域列表中包含注册区域数的区域列表记录计数器设为无效。对区域列表中记录计数器没有激活的记录，启动计数器且持续时间由区域计数器长度描述。
- 如果区域列表包含多于注册区域保留数，根据描述的原则删除多余记录。
- 将列表中不存在的 SID 和 NID 储存值增加到 SID-NID 列表中。
- 使 SID-NID 列表记录计数器失效，对记录计数器没有激活的 SID-NID 列表的其他记录，将计数器按描述的持续时间激活。
- 如果 SID-NID 列表包含多于 N_{10m} = 4 条的记录，则根据基于区域注册的指定原则删除多余记录。N_{10m} 表示 SID-NID 列表大小。
- 如果多个 SID 储存标识等于 0 而且 SID-NID 列表包含有不同 SID 的记录，则根据基于区域注册中指定的原则删除多余记录。
- 如果多个 NID 储存标识等于 0 而且注册 SID-NID 列表中对一个 SID 包含多个记录，则根据基于区域注册中指定的原则删除多余记录。
- 将上次注册距离地点的储存值设为基站地点，将注册距离的储存值设为基站注册距离。
- 更新漫游状态并按漫游类型描述设置移动台终止标识。移动台应指示用户是否漫游。

10. 邻近列表更新消息

移动台支持大小至少为 N_{8m} = 20 个导频的邻近组。在移动台首次指配前向业务信道时初始化邻近组，让其包括最近接收的邻近列表消息中描述的导频。移动台为邻近组中所有导频保持计数器（邻近列表时限）。在移动台将导频从激活组或候选组移到邻近组时，将计数器初始化为 0。在移动台将导频从剩余组移到邻近组，将计数器初始化为邻近组保持的最长时限。一旦接收到邻近列表更新消息，移动台将邻近组中导频的计数器增 1。

如果移动台接收邻近列表更新消息，执行如下操作：

- 将邻近组中所有导频的邻近列表时限增 1。
- 从邻近组中删除所有计数器超过邻近组最长时限的导频。
- 向邻近组中添加消息中提到的且不属于候选组和剩余组中的导频。如果移动台在邻近组中只能储存 k 个导频但在邻近列表更新消息中有多于 k 个导频，移动台就储存消息中前 k 个导频。

11. 功率控制参数消息

移动台储存功率控制参数消息中的如下参数：

- 功率控制报告门限
- 功率控制报告数
- 门限报告模式标识
- 周期报告模式标识
- 功率报告时延

移动台将接收的帧总数以及接收的坏帧总数设为 0。

12. 释放指令

移动台使发送器失效。如果移动台通过关机标识进入释放指令，移动台关机。否则，移动台进入初始化状态的系统检测子状态。

13. 获取参数消息

移动台在 T_{56m} 秒内发送参数响应消息。

14. 服务选项控制指令

如果当前有一激活服务选项(即，激活服务选项数不等于 NULL)，移动台依照激活服务选项要求解释服务选项控制指令的消息行动时间并按照要求处理服务选项控制指令。否则，移动台在 T_{56m} 秒内发送移动台拒绝指令 00000001。

15. 状态请求指令

移动台在 T_{56m} 秒内发送状态消息。

如果移动台接收其他消息，其消息类型在前向业务信道消息体格式中描述，移动台处理消息的所有 2 层字段。如果移动台接收的消息不在以上所列或无法处理，移动台丢弃消息并在 $T_{56m}=0.2$ 秒内发送拒绝指令。T_{56m} 表示响应前向业务信道上所接收的消息或指令的默认最长时间。

第 6 章　基站呼叫处理

本章介绍基于 TIA/EIA/IS-95 的基站呼叫处理。呼叫处理是指基站和移动台之间的消息流协议。

基站呼叫处理有以下几种类型：

- 导频和同步信道处理
  在此期间，基站发射导频和同步信号，使移动台捕获和同步到 CDMA 信道。同时移动台处于初始化状态。
- 寻呼信道处理
  在此期间，基站发射寻呼信号，使移动台监视器可接收消息。同时移动台处于空闲状态，或系统接入状态。
- 接入信道处理
  在此期间，基站监视接入信道，以接收移动台发来的消息。同时移动台处于系统接入状态。
- 业务信道处理
  在此期间，基站用前向业务信道和反向业务信道与移动台交换信息。同时移动台处于业务信道控制状态。

6.1　导频和同步信道处理

1. 导频信道是移动台用于捕获信道和定时的参考信道，并作为相干解调的相位参考。对基站支持的各 CDMA 信道，基站都会不断地发射导频信号。
2. 同步信道为移动台提供系统配置和定时信息。对基站支持的每一个 CDMA 信道，基站最多有一个同步信道。如果基站支持主 CDMA 信道，就会为其发射一个同步信号。如果基站不支持主 CDMA 信道，就会为次级 CDMA 信道发射一个同步信号。基站会不断地为其各同步信道发射同步信道消息。

6.2　寻呼信道处理

基站发射寻呼信号，使移动台监视器可接收消息。同时移动台处于空闲状态，或系统接入状态。对每个所支持的 CDMA 信道，基站具有多至 7 个寻呼信道。对基站为其发射了同步信号的各 CDMA 信道，基站都会再发射至少一个寻呼信号。

对基站发射的各寻呼信号，基站都会不断发射有效寻呼信道消息，其中可包括零消息。基站不会发送任何以寻呼信道时隙为终点的消息，除非该寻呼信道时隙是消息的起点，或该寻呼信道时隙的前一个寻呼信道时隙是消息的起点。

6.2.1 寻呼信道处理

1. 输入以下两个数据，基站就可用与移动台相同的 Hash 函数，从 N 个可用的资源中选择一个，以确定移动台的指配 CDMA 信道：
   - 移动台识别号（MIN）
   - 基站为其发射寻呼信号的 CDMA 信道号
2. 输入以下两个数据，基站就可用 Hash 函数来确定移动台的指配寻呼信道：
   - 移动台识别号（MIN）
   - 基站发射给 CDMA 信道的寻呼信道号
3. 输入以下两个数据，基站即从 Hash 函数的结果中选择一个数，以确定在给定循环指数 i 下，移动台的指配寻呼信道时隙：
   - 移动台识别号（MIN）
   - 寻呼信道最大时隙数（2 048）

移动台指配寻呼信道时隙（$\lfloor t/4 \rfloor$——用来确定寻呼信道时隙的 Hash 值）就是那些满足 mod（$16\times T$）= 0 的时隙。其中 t 为帧中的系统时间，T 为时隙周期长度，以 1.28 秒为单位，由 $T = 2^i$ 确定，其中 i 为时隙周期指数。

6.2.2 消息发送和确认过程

有了寻呼信道确认过程，基站和移动台之间在寻呼信道和接入信道上的信息交换变得可靠。

基站使用确认地址类型，确认序列号，消息序列号，要求的确认，有效确认等字段，以支持这种机制。这些字段可参考第 2 层的字段，确认过程也可参考第 2 层的过程。有关的其他所有字段和处理过程可参考第 3 层相应的部分。

可通过地址字段将寻呼信道消息接入具体的移动台，也可将其接入具体的 MIN（只适用于寻呼消息和时隙化的寻呼消息）。由于 MIN 在多个移动台都有效，因此每种消息地址类型都有各自的确认过程和消息序列编号过程。

对于每种消息地址类型（即地址类型字段的每个允许值）和每种地址的寻呼信道，基站都给出独立的消息编号序列。消息和时隙化寻呼消息的记录是用 MIN 来寻址的（就好像地址类型等于 000 的情况）。

对于每个消息地址类型，需要确认的消息和不需确认的消息都有各自的消息编号序列。一个基站的序列号可以是独立于其他基站的。对于发向消息地址的新消息，基站会将相应的消息序列号的值加 1（模 8）。

基站发射消息序列号（包含在发向一个消息地址的消息中）之后，至少会等待 T_{4m} = 2.2 秒，才可以在其他消息中使用相同的消息序列号。

为增加消息接收的概率，基站可多次发射一个消息。基站能在首次发射消息后的 T_{4m} 秒之内完成同一消息的多次发射，如图 6.1 所示。首次发射后，如果基站在大于 T_{4m} 秒的时间内才发射完包含相同内容的消息，则会用到不同的消息序列号。

为确认最新收到的来自移动台的接入信道消息，基站会将有效确认指示符字段设置为 1。当收到有效确认字段为 1 的消息时，基站会用收到的确认地址类型、确认序列号和移动台识别字段来

对其进行检测。如果已经收到了消息的确认，基站就不能再发射"要求确认"的消息。

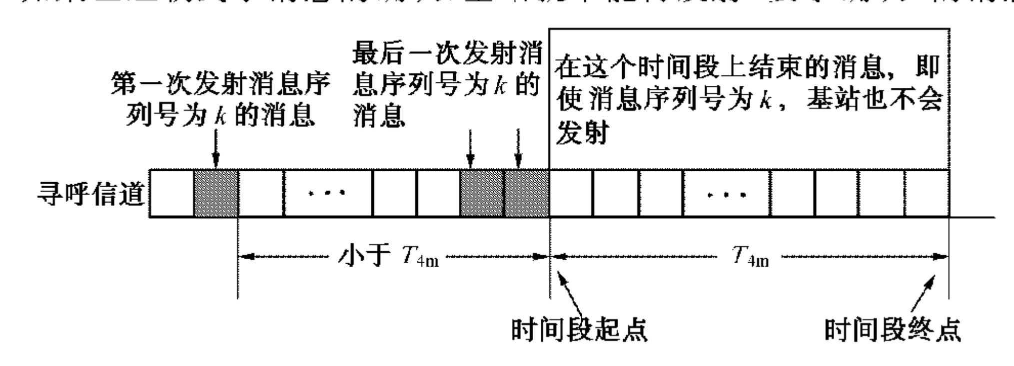


图 6.1　消息序列号再用

6.2.3　开销消息

基站发送开销消息，为移动台提供与基站共同工作所需的信息。

基站给出配置序列号，并且只要基站修改了系统参数消息、相邻列表消息或 CDMA 信道列表消息，就将配置序列号加 1（模 64）。

基站给出接入配置序列号，并且只要基站修改了接入参数消息，就将接入配置序列号加 1（模 64）。

每 T_{1b} = 1.28 秒，基站至少为其发射的每个寻呼信道发送一次下列系统开销消息。T_{1b}表示基站为寻呼信道发射开销消息的最大周期。

1. 接入参数消息

   接入参数消息定义了移动台通过接入信道向基站发射消息时所用的参数。基站发送接入参数消息，要用以下变长消息格式：

   - 消息类型（8 比特）
     基站将该字段设置为 00000010。
   - 导频 PN 码序列偏移指示（9 比特）
     基站将该字段设置为其导频 PN 码序列偏移（以 64 个 PN 码片为单位）。
   - 接入参数消息序列号（6 比特）
     基站将该字段设置为接入配置序列号。
   - 接入信道数（5 比特）
     基站将该字段设置为比该寻呼信道对应的接入信道数少 1。
   - 额定发射功率偏移（4 比特）
     基站将该字段设置为移动台在开环功率估计中所用的校正因子，用两位的补码表示，以 1 dB 为单位。
   - 接入初始功率偏移（5 比特）
     基站将该字段设置为移动台在接入信道上初始发射时，开环功率估计中所用的校正因子，用两位的补码表示（以 1 dB 为单位）。
   - 功率增量（3 比特）
     基站将该字段设置为接入探测序列中逐次发射的接入探测间的发射功率增量，以1 dB为单位。

- 接入探测数(4 比特)
  基站将该字段设置为移动台准备发射的单个接入探测序列中接入探测的最大数目减 1。
- 接入信道消息封装的最大长度(3 比特)
  基站将该字段设置为 0 到 7 之间的一个值,比接入信道消息封装中接入信道帧的最大数目少 3。
- 接入信道报头长度(4 比特)
  基站将该字段设置为移动台准备在各接入信道报头中发射的接入信道帧的数目减 1。
- 0 ~ 9 级接入过载的暂留值(6 比特)
  如果允许移动台在 0 ~ 9 级接入过载下在接入信道上发射请求信号,基站就会将该字段设置为将用到的暂留值。否则,基站就会将该字段设置为 111111。
- 10 级接入过载的暂留值(测试移动台)(3 比特)
  如果允许移动台在 10 级接入过载下在接入信道上发射请求信号,基站就会将该字段设置为将用到的暂留值。否则,基站就会将该字段设置为 111。
- 11 级接入过载的暂留值(移动台电平)(3 比特)
  如果允许移动台在 11 级接入过载下在接入信道上发射请求信号,基站就会将该字段设置为即将用到的暂留值。否则,基站就会将该字段设置为 111。
- 12,13,14 和 15 级接入过载的暂留值(均为 3 比特)
  如果允许移动台在 12,13,14 或 15 级接入过载下在接入信道上发射请求信号,基站就会将该字段设置为即将用到的暂留值。否则,基站就会将该字段分别设置为 111。
- 尝试发射消息的接入信道暂留修正量(3 比特):α
  移动台的发射概率以 $2^{-\alpha}$ 数量级增加。基站将该字段设置为尝试发射消息的接入信道暂留修正量。
- 尝试注册(不是对注册请求指令的响应)的接入信道暂留修正量(3 比特)
  移动台的发射概率以 $2^{-(\text{REG-PSIST})}$ 数量级增加。基站将该字段设置为尝试注册(不是对注册请求指令的响应)的接入信道暂留修正量。
- 接入信道探测的时间随机化(4 比特):β
  移动台通过 RN 个 PN 码片,使信号时延于系统时间发射,其中 RN 是由 0 到 $2^{-\beta}-1$ 个 PN 码进行 Hash 变换得到的一个数,也就是说,由移动台生成(伪随机)的 PN 码片的时延优先于进行接入尝试。基站将该字段设置为 0 ~ 9 之间的一个数,因此时间的随机化范围为 $2^{-\beta}-1$ 个 PN 码片。
- 接入信道确认超时(4 比特)
  移动台在接入信道上发射完消息后,需等待一段时间才能确定基站是否收到了信号。基站就把"接入信道确认超时"这个字段设置为移动台等待时间减 2(以 80 ms 为单位)。
- 接入信道探测补偿范围(4 比特)
  基站将该字段设置为由于逐次接入探测间的随机补偿引起的移动台时延的最大时隙数减 1。
- 接入信道探测序列补偿范围(4 比特)
  基站将该字段设置为由于后续接入探测序列间的随机补偿引起的移动台时延的最大时隙数减 1,并且要在"响应接入"信号的第一个接入探测序列之前。

- 接入信道请求的最大接入探测序列数(4 比特)

  基站将该字段设置为移动台将为接入信道请求发射的接入探测序列的最大数目。该值必须大于 0。
- 接入信道响应的最大接入探测序列数(4 比特)

  基站将该字段设置为移动台将为接入信道响应发射的接入探测序列的最大数目。该值必须大于 0。
- 鉴权模式(2 比特)

  如果移动台准备将标准鉴权资料包含在接入信道消息中,基站就将该字段设置为 01,否则基站就将该字段设置为 00。其他值都作为预留。
- 随机查询值(0 或 32 比特)

  如果鉴权模式字段设置为 01,基站就会将"随机查询值"字段设置为移动台鉴权所用的随机查询值。否则,基站就会忽略掉该字段。
- 保留比特(7 比特)

  基站永远将该字段设置为 0000000。

2. CDMA 信道列表消息

基站用下列变长消息格式发送 CDMA 信道列表消息。

- 消息类型(8 比特)

  基站将该字段设置为 00000100。
- 导频 PN 码序列偏移指示(9 比特)

  基站将该字段设置为其导频 PN 码序列偏移(以 64 个 PN 码片为单位)。
- 配置消息序列号(6 比特)

  基站将该字段设置为配置消息序列号。
- CDMA 信道频段(11 比特)

  包含该字段取值的指令指定了该字段所支持的 CDMA 信道,从 CDMA 信道 1 到 CDMA 信道 n。

  对每个包含了基站所支持的寻呼信道的 CDMA 信道,基站都相应的有一个该字段的取值。如果基站支持主 CDMA 信道,它就会首先给出主 CDMA 信道对应的该字段值。否则,如果基站支持的是次级 CDMA 信道,则基站会首先给出次级 CDMA 信道对应的该字段值。

  基站将该字段值设置为 CDMA 信道号,该信道号对应于 CDMA 信道频段。
- 保留比特(0 ~ 7 比特,依需要而定)

  基站依需要加入保留比特,以使使整个消息的长度等于一个八位的整数。基站把保留比特设置为 0。

3. 相邻列表消息

基站用下列变长消息格式发送相邻列表消息。

- 消息类型(8 比特)

  基站将该字段设置为 00000011。
- 导频 PN 码序列偏移指示(9 比特)

基站将该字段设置为其导频 PN 码序列偏移(以 64 个 PN 码片为单位)。

- 配置消息序列号(6 比特)

  基站将该字段设置为配置消息序列号。

- 导频 PN 码序列偏移指示增量(4 比特)

  移动台在导频 PN 码序列指示值(为该增量的倍数)中搜索剩余集导频。基站将该字段设置为导频序列增量(以 64 个码片为单位),移动台将用该增量来搜索剩余集。基站必须将该字段设置为最大增量,这样才能使所有相邻基站的导频 PN 码序列偏移都是该增量的整数倍。

  基站包含了下面两个字段的记录的一个取值。记录中每一项都会置于移动台的邻近集里。基站可包含该记录的 0 个或者更多的取值。

- 相邻配置(3 比特)

  基站将该字段设置为表 6.1 所示"相邻配置"项的值。

表 6.1 相邻配置字段

字段值	相邻配置
000	相邻基站与当前基站配置相同
001	相邻基站与当前基站配置不同。相邻基站在当前 CDMA 频段上没有主寻呼信道
010	相邻基站在当前 CDMA 频段上没有寻呼信道。在当前基站发射的 CDMA 信道列表消息中列出的第一个 CDMA 频段上,相邻基站有一个主寻呼信道
011	相邻基站配置未知
100 ~ 111	预留值

- 相邻导频 PN 码序列偏移指示(9 比特)

  基站将该字段设置为相邻信号的导频 PN 码序列偏移,以 64 个 PN 码片为单位。

- 保留比特(0 ~ 7 比特,依需要而定)

  基站依需要加入保留比特,以使使整个消息的长度等于一个八位的整数。基站把保留比特设置为 0。

4. 系统参数消息

基站用下列定长的消息格式发送系统参数消息。

- 消息类型(8 比特)

  基站将该字段设置为 00000011。

- 导频 PN 码序列偏移指数(9 比特)

  基站将该字段设置为其导频 PN 码序列偏移(以 64 个 PN 码片为单位)。

- 配置消息序列号(6 比特)

  基站将该字段设置为配置消息序列号。

- 系统标识(SID)(16 比特)

  该字段是由系统所有者定义的系统子标识符。

  基站将该字段设置为网络识别号。

  网络识别号 65 535 为预留值。

- 注册区域(12 比特)

基站将该字段设置为注册区域号。

- 保留注册区域数(3 比特)

  基站将该字段设置为移动台为了进行基于区域的注册而保留的注册区域数。如果不能进行基于区域的注册,基站就将该字段设置为 000。

- 区域计时器长度(3 比特)

  基站将该字段设置为表 6.2 所示的区域计时器长度值,对应于移动台所用的区域注册计时器长度。

表 6.2　区域计时器值

区域计时器值(二进制)	计时器长度(分)
000	1
001	2
010	5
011	10
100	20
101	30
110	45
111	60

- 多 SID 存储标识符(1 比特)

  如果移动台能够存储包含不同 SID 的 SID-NID 列表的条目,基站就将该字段设置为 1。否则设置为 0。

- 多 NID 存储标识符(1 比特)

  如果移动台能够存储同一 SID(不同 NID)的 SID-NID 列表的多个条目,基站就将该字段设置为 1,否则设置为 0。

- 基站标识(16 比特)

  基站将该字段设置为其识别号。

- 基站级别(4 比特)

  基站将公共宏蜂窝系统的这个字段设置为 0000。其他值都作为预留。

- 寻呼信道号(3 比特)

  基站将该字段设置为此 CDMA 信道上的寻呼信道号。该字段不能为 000。

- 最大时隙周期指数(3 比特)

  基站将该字段设置为最大允许时隙周期长度对应的时隙周期指数值。

- 本地注册标识符(1 比特)

  如果移动台(非漫游且移动台终止标识符等于 1)能够自主注册,基站就会将该字段设置为 1,否则设置为 0。

- SID 漫游注册标识符(1 比特)

  如果移动台(外来漫游者且移动台 SID 终止标识符等于 1)能够自主注册,基站就会将该字段设置为 1,否则设置为 0。

- NID 漫游注册标识符(1 比特)

  如果移动台(外来漫游者且移动台 NID 终止标识符等于 1)能够自主注册,基站就会将该

字段设置为 1,否则设置为 0。

- 开电源注册标识符(1 比特)

  如果(有自主注册能力的)移动台打开电源或收到系统开销消息后要立刻注册,基站就将该字段设置为 1,否则设置为 0。

- 关电源注册标识符(1 比特)

  如果(有自主注册能力的)移动台关掉电源后要立刻注册,基站就将该字段设置为 1,否则设置为 0。

- 参数改变注册标识符(1 比特)

  如果移动台改变参数时要进行注册,基站就将该字段设置为 1,否则设置为 0。

- 周期注册(7 比特)

  如果移动台不进行基于计时器的注册,基站就将该字段设置为 0000000。否则就设置为 29 ~ 85 之间的一个值。这样,期望的计时器值就是:

$$\lfloor 2^{(Reg-Prd)/4} \rfloor \times 0.08 \text{ 秒}$$

- 基站纬度(22 比特)

  基站将该字段设置为基站的纬度,以 0.25 秒为单位,用两位的补码表示,正数代表北极。

- 基站经度(23 比特)

  基站将该字段设置为基站的经度,以 0.25 秒为单位,用两位的补码表示,正数代表东经。

- 注册距离(11 比特)

  如果移动台要进行基于距离的注册,基站就将该字段设置为一个非零的距离值,超过这个距离,移动台就要进行注册。如果移动台不进行基于距离的注册,基站就将该字段设置 0。

- 激活集和候选集的搜索窗大小(4 比特)

  基站将该字段设置为表 5.1 所示移动台用于激活集和候选集的"搜索窗大小"值。

- 邻近集的搜索窗大小(4 比特)

  基站将该字段设置为表 5.1 所示移动台用于邻近集的"搜索窗大小"值。

- 邻近集的搜索窗大小(4 比特)

  基站将该字段设置为表 5.1 所示移动台用于剩余集的"搜索窗大小"值。

- 邻近集最大寿命(4 比特)

  基站将该字段设置为邻近集最大寿命值,超过这个值,移动台就会从邻近集中撤销项。

- 功率控制报告门限(5 比特)

  基站将该字段设置为一个测量周期中收到的坏帧数,之后移动台才产生功率测量报告消息。如果基站将门限报告模式标识符设置为 1,则门限值一定不能设置为 00000。

- 功率控制报告帧数(4 比特):γ

  帧数超过某个值时,移动台要开始计坏帧数。基站设置的"功率控制报告帧数"字段应当使

$$\lfloor 2^{\gamma/2} \times 5 \text{ 帧} \rfloor$$

  的得数等于该值。

- 门限报告模式标识符(1 比特)

  如果移动台要产生门限功率测量报告消息,基站就将该字段设置为 1,否则设置为 0。

- 周期功率报告模式标识符(1 比特)

  如果移动台要产生周期功率测量报告消息,基站就将该字段设置为 1,否则设置为 0。
- 功率报告时延(5 比特)

  移动台在等待周期之后发射功率测量报告消息,然后重新开始计帧数,以达到功率控制的目的。这个等待周期就是由"功率报告时延"字段来设定的(以 4 帧为单位)。
- 再扫描标识符(1 比特)

  如果移动台要通过接收这个消息对系统进行再初始化和再捕获,基站就将该字段设置为 1。
- 导频检测门限(6 比特)

  移动台用该值来触发导频强度测量消息的发送,以启动切换过程。基站将该字段设置为导频检测门限,用一个无符号的二进制数来表示,其值等于$\lfloor -2\times 10\times \log_{10} E_c/I_0 \rfloor$。
- 导频撤销门限(6 比特)

  移动台用该值来起动激活集和候选集中导频的切换撤销计时器。基站将该字段设置为导频撤销门限,用一个无符号的二进制数来表示,其值等于$\lfloor -2\times 10\times \log_{10} E_c/I_0 \rfloor$。
- 激活集对候选集的比较门限(4 比特)

  当候选集中的导频强度超过激活集中的导频强度,移动台会根据差值发射导频强度测量消息。基站将该字段设置为门限候选集导频和门限激活集导频的比值,以0.5 dB为单位。
- 撤销计时器值(4 比特)

  计时器达到该值后,移动台会对导频(激活集或候选集中的项,其强度不大于撤销计时器的值)采取操作。如果该导频是激活集中的项,移动台就发布导频强度测量消息。如果该导频是候选集中的项,移动台就将其移入邻近集。基站将该字段设置为表 6.3 所示的撤销计时器截止值,对应于移动台所用的撤销计时器值。

表 6.3　切换撤销计时器截止值

撤销计时器值	计时器截止值(秒)	撤销计时器值	计时器截止值(秒)
0	0.1	8	27
1	1	9	39
2	2	10	55
3	4	11	79
4	6	12	112
5	9	13	159
6	13	14	225
7	19	15	319

  移动台在发射导频强度测量消息时,会指明激活集和候选集中所有导频的切换撤销计时器状态。
- 保留比特(4 比特)

  基站将该字段设置为 0000。

6.2.4　面向移动台的消息

基站选择寻呼信道时隙来向移动台发送消息时,遵循以下规则:

- 如果基站能够确定移动台是在非时隙化模式下运行,基站就可在任何寻呼信道时隙上发送

消息给移动台。

- 如果基站能够确定移动台是在时隙化模式下运行，也能确定移动台时隙周期指数，基站就会以如下方式发送至少一次消息：

  1. 基站在指配寻呼信道时隙中向移动台发送消息；
  2. 在最近一次发送时隙化寻呼消息之后，基站不会再用同一寻呼信道时隙发送消息。

- 如果基站不能确定移动台是在时隙化模式下运行，或者不能确定移动台时隙周期指数，基站会假定移动台是在时隙化模式下运行，其时隙周期指数为最大时隙周期指数和 1 中较大的那个。基站会以如下方式发送至少一次消息：

  1. 基站在指配寻呼信道时隙中向移动台发送消息；
  2. 在最近一次发送时隙化寻呼消息之后，基站不会再用同一寻呼信道时隙发送消息。

基站在每个寻呼信道时隙中至少发送一个时隙化寻呼消息。基站会将直接发送给移动台(运行在时隙化模式下)的消息作为时隙中首选的消息。

基站可在寻呼信道中直接发送下列消息给移动台。基站在发送消息时必须遵守一些特殊要求：

1. 缩写的提醒指令
2. 监察指令
3. 鉴权查询消息
4. 基站确认指令
5. 基站查询确认指令
6. 信道指配消息
7. 数据突发消息
8. 特征通知消息
9. 插入指令
10. 本地控制指令
11. 锁定指令
12. 要求维护指令
13. 寻呼消息：当寻呼外来 SID 漫游者或外来 NID 漫游者时，基站发送的消息要包含 MIN1 和 MIN2 两个字段。
14. 接受注册指令
15. 拒绝注册指令
16. 注册请求指令
17. 释放指令
18. 重排序指令
19. 时隙化寻呼消息：当寻呼外来 SID 漫游者或外来 NID 漫游者时，基站发送的消息要包含 MIN1 和 MIN2 两个字段。
20. SSD(站址选择分集)更新消息
21. 解锁指令

6.3　接入信道处理

在此期间，基站监视接入信道以接收移动台在系统接入状态时发送的消息。

每个接入信道都对应于一个寻呼信道。和同一个寻呼信道相对应的接入信道可多至 32 个。基站发送给寻呼信道的接入参数消息中指定了和具体的寻呼信道相对应的接入信道数。基站不间断地监视与其发射的寻呼信道相对应的所有接入信道。

6.3.1　接入信道确认过程

有了接入信道确认过程，基站和移动台之间在寻呼信道和接入信道上的信息交换变得可靠。基站使用确认地址类型，确认序列号，消息序列号，要求确认标识符，有效确认等字段，以支持这种机制。这些字段可参考第 2 层的字段，确认过程也可参考第 2 层的过程。有关的其他所有字段和处理过程可参考第 3 层相应的部分。

如果要求确认字段设置为 1，接入信道收到的消息就需要确认。在本书中，接入信道上发送的所有消息都需要确认。接入信道上发送的所有消息包含了移动台发送消息的识别数据，并由寻呼信道消息来确认。

基站通过在寻呼信道上发射消息来确认收到的消息，其中确认序列号字段设置为与收到的消息的消息序列号相等，且有效确认字段设置为 1。以这种方式设置“确认序列号”和“有效确认”字段被称做“包含被接收消息的一个确认”。

在接入信道上收到移动台发来的需要确认的消息之后，基站会在相应的寻呼信道上直接向移动台发射消息，消息中包含确认。确认是在接入信道确认超时之内发射的，以 80 ms 为单位。其中接入信道确认超时的值在接入参数消息中。接入参数消息是在移动台指配寻呼信道上发送。

如果收到消息后，被接收消息需要确认，并且在接入信道确认超时 ×80 ms 内没有消息直接发送给移动台，基站就会直接给移动台发射一个基站确认指令，其中包含确认。

无论何时，只要收到来自移动台的需要确认的消息，基站随后都会直接向被接收消息中指定的移动台消息序列号发射寻呼信号消息，并在其中设置确认序列号。在此确认序列号下发射的第一个消息，其“有效确认”字段要设置为 1。第一个消息之后的每一个寻呼信道消息都直接发射给同一个移动台，并包含相同的确认序列号：

- 如果消息是在第一个消息之后 T_{4m} 秒内发送的，基站就将“有效确认”字段设置为 1。
- 如果消息是在第一个消息之后大于 T_{4m} 秒发送的，基站就将“有效确认”字段设置为 0。

基站要用接入信道消息序列号进行复制消息检测，应当采取如下过程。对每一个在接入信道上处于激活状态的移动台，基站都应为接入信道消息序列号字段[n]的每一个可能的值存储一个接收状态标识符，这里的 n 介于 0 ~ 7 之间。

基站在收到来自移动台的接入信道消息时，应认为该移动台在接入信道上是处于激活状态的。如果出现下列情况中的一种，基站应认为该移动台在接入信道上处于非激活状态：

- 在一个时间段内，基站没有收到来自移动台的任何消息。这个时间段是由基站制造商来选择的。
- 移动台已经被指配为业务信道。

- 移动台已经被指配为模拟系统。
- 基站已经收到来自移动台的关电源注册。

如果收到来自非激活状态移动台的接入信道消息,基站应当将收到的消息序列号[n]设置为NO,无论收到的是0~7之间的哪个数。之后基站应认为该移动台在接入信道上已经处于激活状态。

对于每个处于激活状态的移动台,基站应当采取下列过程:

- 如果基站收到一个需要确认并带有"消息序列号"的消息(包括移动台处于非激活状态时基站收到的一个消息),并且收到的消息序列号为NO,基站应当将收到的这个消息作为一个新消息来处理。如果基站收到的消息序列号为YES,基站就应将收到的这个消息序列号设置为"消息序列号"+2(模8)后等于NO。
- 如果收到一个需要确认并带有消息序列号的消息,同时收到的消息序列号为YES,基站会按规定在这个部分早些确认该消息,但不会再对该消息进行进一步的处理。

6.3.2 寻呼响应消息、发起消息和注册消息的响应

1. 如果收到寻呼响应消息,基站应发送信道指配消息或释放指令,也可开始鉴权过程。
   如果基站发送的是信道指配消息,就要执行如下操作:
   - 如果信道指配消息将移动台定向到业务信道上,基站就开始业务信道处理过程。
   - 如果信道指配消息将移动台定向到模拟话音信道上,基站就采取对应的过程。
2. 如果收到发起消息,基站应发送信道指配消息,插入指令,重排序指令,或者释放指令,也可开始鉴权过程。
   如果基站发送的是信道指配消息,就要执行如下操作:
   - 如果信道指配消息将移动台定向到业务信道上,基站就开始业务信道处理过程。
   - 如果信道指配消息将移动台定向到模拟话音信道上,基站就采取对应的过程。
3. 如果收到注册消息,基站可发送接受注册指令或拒绝注册指令。基站也可开始鉴权过程。鉴权过程用于在移动台和基站之间进行信息交流,以证实移动台身份。只有证明移动台和基站拥有同一套共享加密数据时,鉴权才会成功。

6.4 业务信道处理

在此期间,基站用前向业务信道和反向业务信道与移动台交换信息。同时移动台处于业务信道控制状态。

业务信道处理由下列子状态组成:

- 业务信道初始化子状态
  在此子状态,基站开始在前向业务信道上发射信号,在反向业务信道上接收信号。
- 等待指令子状态
  在此子状态,基站向移动台发送信息示警消息。
- 等待应答子状态

在此子状态，基站等待来自移动台的接通指令。

- 会话子状态

  在此子状态，基站与移动台主服务选择应用交换主业务比特。

- 释放子状态

  在此子状态，基站切断呼叫。

接下来将介绍基站在各个业务信道处理子状态中所进行的特定的功能和操作。

6.4.1 前向业务信道功率控制

如果基站启用了前向业务信道功率控制，移动台就用功率测量报告消息向基站报告帧差错率统计数字。基站启用前向业务信道功率控制的方法在寻呼信道上发送系统参数消息并在前向业务信道上发送功率控制参数消息。基站可启用周期报告，使移动台在指定的时间间隔定期向基站报告帧差错率统计数字。基站也可启用门限报告，当帧差错率达到规定门限时，移动台就会向基站报告帧差错率统计数字。基站根据移动台报告的帧差错率统计数字来调整前向业务信道发射功率。

6.4.2 服务选择

业务信道工作期间，基站和移动台可支持主业务服务。基站和移动台处理前向信道和反向信道主业务比特的通道就是由主业务服务用一套规定来管理的。

基站和移动台都可以申请服务选择。基站在寻呼移动台或在业务信道工作期间，都可申请特殊的服务选择。如果基站的服务选择申请是移动台可以接受的，则基站和移动台就开始使用新的服务选择。如果基站申请的服务选择是移动台不能接受的，移动台可以拒绝这次服务选择申请，或申请一个待选服务选择。如果移动台申请了一个待选服务选择，基站可以接受也可以拒绝该方案，或再申请另一个服务选择。这个过程就叫做服务选择协商。当基站和移动台找到了一个双方都接受的服务选择，基站拒绝了移动台的服务选择申请，或移动台拒绝了基站的服务选择申请时，这种协商才会结束。

基站和移动台用“服务选择申请指令”来申请服务选择或提出待选服务选择，用“服务选择响应指令”来接受或拒绝服务选择申请。此外，基站还可以在寻呼消息或时隙化寻呼消息中申请服务选择，移动台也可以在发起消息或寻呼响应消息中申请服务选择。基站和移动台用服务选择控制指令来调用服务选择的特殊功能。

在服务选择申请指令中，基站用一个可变的“服务选择申请数”来记录服务选择数，基站已为这些服务选择发送了待处理申请。如果基站没有发送待处理的服务选择申请，服务选择申请数就设置为一个特殊值 NULL。基站用另一个可变的值来记录当前处于激活状态的服务选择数。如果没有处于激活状态的服务选择，当前服务选择数就设置为 NULL。

6.4.3 服务选择处理

1. 服务选择申请处理

   基站在发起消息、寻呼响应消息或服务选择申请指令中处理服务选择申请时，要执行下列操作：

   - 如果基站接受了移动台所申请的服务选择，就会将服务选择申请数设置为 NULL，并在 T_{4b} =

5 秒内发送一个“接受所申请服务选择”的服务选择响应指令。T_{4b} 为基站响应一个服务选择申请的最长时间。基站按照服务选择要求开始使用移动台所申请的服务选择。服务选择激活后，基站就将当前服务选择设置为移动台所申请的服务选择数。

- 如果基站没有接受移动台所申请的服务选择，并且有一个可申请的待选服务选择，基站就会将服务选择申请数设置为该待选服务选择数，并在 T_{4b} 秒内发送一个服务选择申请指令以申请待选服务选择。
- 如果基站没有接受移动台所申请的服务选择，也没有可申请的待选服务选择，基站就会将服务选择申请数设置为 NULL，并在 T_{4b} 秒内发送一个服务选择申请指令以拒绝该申请。基站会和先前一样继续处理主业务，而后才会再接受服务选择申请指令，同时基站仍然处于目前的状态。

2. 服务选择响应指令处理

如果收到服务选择响应指令，基站会执行下列操作：

- 如果该指令中指定的服务选择数等于服务选择申请数，基站就会将服务选择申请数设置为 NULL，并按照服务选择要求开始使用移动台所申请的服务选择。服务选择激活后，基站就将当前服务选择设置为移动台所指定的服务选择数。
- 如果该指令表明服务选择申请被拒绝，基站就会将服务选择申请数设置为 NULL。接着基站和先前一样继续处理主业务，而后才会再接受服务选择申请指令，同时基站仍然处于目前的状态。
- 如果该指令没有表明服务选择申请被拒绝，指令中指定的服务选择数也不等于服务选择申请数，基站就会将服务选择申请数设置为 NULL，并发送释放指令 00000010，进入释放子状态。

3. 对收到的服务选择控制指令的处理

如果有一个服务选择是处于激活状态的(即其当前服务选择数不为 NULL)，基站就会按照该服务选择的要求来处理收到的服务选择控制指令。

4. 服务选择申请初始化

要进行服务选择申请初始化，基站需将服务选择申请数设置为指定的服务选择数。

6.4.4 确认过程

确认过程使基站和移动台之间在寻呼信道和接入信道上的信息交换变得可靠。基站使用确认序列号、消息序列号、要求确认等字段，以检测复制的消息，并提供确认过程的一个参考模型。这些字段可参考第 2 层的字段，确认过程也可参考第 2 层的过程。其他所有消息字段可参考第 3 层的字段，处理过程可参考第 3 层的处理过程。

在反向业务信道和前向业务信道上，消息确认过程是一个选择性的重复过程。在此过程中，只有当没有收到消息的确认时，才会重发消息。

1. 发送消息与接收确认

如果要求确认字段为 1，业务信道消息就需要确认。第二层协议并不保证消息以任何顺序传输。如果基站要使移动台以某种顺序接收一组消息，基站就必须在等到每个消息的确认后再发射下一个消息。如果需要确认的消息之间的相对顺序并不重要，基站可在收到第一

个消息的确认之前发射最多 4 个消息。

基站为需要确认的消息存储了一个消息序列号。对前向业务信道消息序列号字段(确认等待值[n],其中 n 在 0～7 之间)的每一个可能值,基站都存储了一个确认状态标识符。如果确认等待值(消息序列确认号 +4,模 8)为 YES,基站不会发送新的需要确认的消息。基站进行下列过程:

- 如果反向业务信道上收到带有确认序列号的消息,基站就将确认等待(消息序列确认号)设置为 NO。
- 如果基站在前向业务信道上发送了新的需要确认的消息,基站就会将确认等待(消息序列确认号)设置为 YES,并将消息序列号字段设置为消息序列确认字段。然后将消息序列确认号加 1(模 8)。

  收到一个消息的确认后,基站不会重发射该消息。

  如果基站在发射消息后没有收到确认,就会重发射该消息。基站重发射消息时会使用和原来相同的消息序列号。

  基站为每一个需要确认的消息存储了一个重发射计数器。在第一次发射某个消息前,基站将重发射计数器设置为 0。如果基站没有收到确认,则每发射一次该消息,重发射计数器的值就加 1。基站的重发射次数不能超过基站制造商选定的最大重发射次数。如果重发射次数达到了最大重发射次数,基站就会宣布确认失败。

2. 接收消息与返回确认

反向业务信道上收到的消息包含的消息序列号字段值增加的规则和前向业务信道上发射的消息相同。需要确认和不需要确认的反向信道消息拥有各自的消息序列号。基站对收到消息的确认是通过发射一条消息来实现的,这条消息中确认序列号字段设置为收到消息的序列号字段。确认序列号设置为这种方式的消息称为"包含(收到消息的)一个确认"。

只要收到一个要求确认的消息,基站就会将随后的反向业务信道消息的确认序列号字段设置为收到消息的消息序列号。如果没有收到消息,基站就会将该字段设置为 111。

收到一个需要确认的消息后,基站会在 T_{1m} = 0.4 秒内发射一个包含确认的消息(如图 5.8 所示)。T_{1m}表示移动台等待确认的最长时间。

如果收到的消息需要确认,并且在收到消息后 T_{1m}秒内没有可接收的消息,基站就会发射一个包含确认的"基站确认指令"。

为检测复制消息,基站会为 RTC 消息序列号字段(即收到的消息序列号[n],其中 n 在 0～7 之间)的每一个可能值存储一个接收状态指示符。基站进行下列过程:

- 如果收到需要确认并带有消息序列号的消息,并且收到的消息序列号为 NO,基站就将该消息作为一个新消息来处理。基站收到的消息序列号设置为 YES,将收到的消息序列号(消息序列号 +4,模 8)设置为 NO。
- 如果收到一个需要确认并带有消息序列号的消息,同时收到的消息序列号为 YES,基站会确认该消息,但不会再对其进行进一步的处理。

3. 不需确认的消息

如果业务信道消息的要求确认标识符字段值为 0,那么它就不需要确认。基站存储了不需要确认消息的消息序列号。对每一个不需要确认的新消息,基站都将消息序列号字段设置

为“无确认消息序列”，然后使“无确认消息”以模 8 增加。

如果基站发射同一个不需确认的消息超过一次，那么每次发射都要采用相同的消息序列号。基站要在第一次发射后的 T_{3m} = 0.32 秒之内完成全部重发射，如图 6.2 所示。基站在上一次发射一个不需确认的消息之后至少要等待 T_{3m} 秒，才会发射另一个带有相同消息序列号的不需确认的消息，如图 6.2 所示。T_{3m} 表明了移动台在前向业务信道上收到两个消息的时间间隔（这两个消息都不需要确认，带有相同的序列号，并被认为是复制的）。

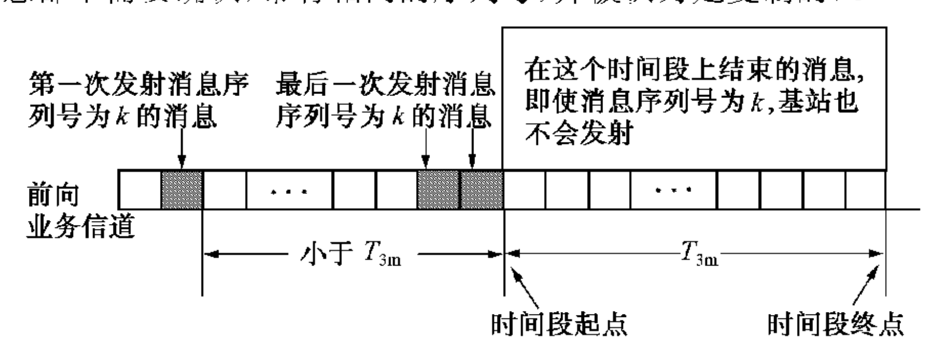


图 6.2 基站不再用某个消息序列号的时间要求

4. 确认过程复位

基站复位确认过程如下：

- 消息序列复位

  对于任何 n，如果确认等待[n]等于 YES，则基站应存储相应的消息，并在重发射消息之前完成确认过程的复位。基站将重发射计数器设置为 0。

- 基站将消息序列确认设置为 0，不需确认的消息序列设置为 0，将确认等待[n]设置为 NO（对于所有在 0 ~ 7 之间的 n）。

- 确认序列号复位

  基站将所有 FTC 消息的确认序列号字段设置为 111，直到收到第一个需要确认的消息为止。

- 复制检测复位

  基站将收到的消息序列号[n]设置为 NO（对于 0 ~ 7 之间的每个 n）。

6.4.5 消息激活次数

若前向业务信道消息不带有“使用时间”字段或“使用时间”字段设置为 0，则消息中会有一个隐含的作用时间。使用时间字段为 1 的消息有一个显性的作用时间，在消息的作用时间字段里定义。带有未来作用时间的消息就叫做“待处理消息”。

除非另有规定，带有隐含作用时间的消息必须在不晚于第一个 80 ms 的时间界限内生效（相对于系统时间），此过程在包含消息最后一个比特的帧结束后至少 80 ms 发生。带有显性作用时间的消息在系统时间（以 80 ms 为单位，模 64）等于消息作用时间字段值时生效。作用时间和包含消息最后一个比特的帧之间的时差至少为 80 ms。

基站在任何给定的时间上都支持待处理消息（不包括待处理服务选择控制指令）。基站需要支持的待处理服务选择控制指令数视具体的服务选择而定。

6.4.6　长码转移请求过程

如果发起消息或寻呼响应消息中请求了话音保密，则基站可发送一个长码转移请求指令 00000001 来请求专用长码的转移。

基站对长码转移请求指令的处理如下：

- 如果长码转移请求指令请求转移专用长码，并且基站接受了请求，则基站发送长码转移请求指令 00000001。如果基站不接受请求，则基站发送长码转移请求指令 00000000。
- 如果长码转移请求指令请求转移公共长码，并且基站接受了请求，则基站发送长码转移请求指令 00000000。如果基站不接受公共长码转移请求，则基站发送长码转移请求指令 00000001。
- 如果长码转移响应指令表明移动台接受了基站发送的长码转移请求指令中所请求的长码转移，基站就将所请求的长码掩码用于前向业务信道和反向业务信道上。如果基站在长码转移请求指令中没有指明显性作用时间，则基站开始使用所请求的长码掩码的时间应在第一个 80 ms 的时间界限内（相对于系统时间起点），并且要在没有收到任何长码转移响应指令的最后一帧后的 $N_{4m} = 20$ 帧之后。

6.5　业务信道初始化子状态

在此子状态，基站开始在前向业务信道上发射消息并捕获反向业务信道。进入业务信道初始化子状态时，基站要进行下列操作：

- 基站将消息确认过程复位。
- 基站将激活服务选择号设置为 NULL 以表明没有处于激活状态的服务选择。
- 基站进行服务选择请求初始化，规定服务选择号为 NULL。
- 基站将前向和反向业务信道长码掩码设置为公共长码掩码。
- 基站将前向和反向业务信道帧偏移设置为移动台指配帧偏移。

处于业务信道初始化子状态时，基站要进行下列操作：

- 基站发射零业务信道数据。
- 基站进行消息确认过程。
- 如果捕获到反向业务信道，基站就会发送基站确认指令。基站应将基站确认指令作为一个需要确认的消息来发送。如果呼叫为以移动台为终点的呼叫，则基站进入等待指令子状态。如果呼叫为移动台始呼呼叫，则基站进入会话子状态。
- 如果没有捕获到反向业务信道，基站要么在寻呼信道上重发信道指配消息并继续处于业务信道初始化子状态，要么停止在前向业务信道上发射消息并放弃移动台业务信道的处理。

6.6　等待指令子状态

在此子状态，基站向移动台发送信息示警消息。

进入业务信道初始化子状态时，基站要进行下列操作：

- 基站处理寻呼响应消息中提出的服务选择申请(见 6.4.2 节)。

处于等待指令子状态时,基站要进行下列操作:

- 基站发射功率控制子信道。
- 如果有一个服务选择处于激活状态(激活服务选择号不等于 NULL),基站就按照激活服务选择的要求来处理收到的主业务比特。否则,基站就丢弃收到的主业务比特。
- 如果有一个服务选择处于激活状态(激活服务选择号不等于 NULL),基站就按照激活服务选择的要求发射主业务比特。否则,基站就发射零业务信道数据。
- 基站进行消息确认过程。
- 如果基站声明反向业务信道连续性出现损耗,它就要向移动台发送释放指令,进入释放子状态。
- 基站可进行前向业务信道功率控制。
- 基站可申请服务选择(见 6.4.2 节)。为此,基站要进行服务选择申请初始化,以指明申请的服务选择号,并发送服务选择申请指令。
- 如果有一个服务选择处于激活状态(激活服务选择号不等于 NULL),基站可发送服务选择指令,以按照激活服务选择的要求调用服务选择特殊功能。
- 基站请求长码转移,可以是自动的,也可以是为了响应(发起消息或寻呼响应消息中所指明的)话音保密请求。
- 基站可进行鉴权过程。
- 基站可发送下列消息。基站发送消息,要遵循以下具体要求:

1. 信息示警消息

   之后基站进入等待应答子状态。

2. 模拟切换导引消息

   之后基站进入等待指令任务状态。

3. 监察指令

   监察指令可在寻呼信道或前向业务信道上发送。

   基站用下列定长格式表示具体指令字段:

   - 指令码(6 比特)

     基站将该字段设置为 000110。

   - 指令限定码(8 比特)

     基站将该字段设置为 00000000。

4. 鉴权查询消息

   基站在寻呼信道上发送鉴权查询消息,要采用下面的定长消息格式:

字　段	长度(比特)

消息类型(00001010)	8
确认序列号	3
消息序列号	3
要求确认标识符	1
有效确认标识符	1
地址类型	3
地址字段长度	4
移动台地址	8 × 地址字段长度
随机查询数据	24
保留比特	3

5. 基站确认指令

基站确认指令可在寻呼信道或前向业务信道上发送。

基站用下列定长格式表示具体指令字段：

- 指令码(6 比特)

  基站将该字段设置为 010000。
- 指令限定码(8 比特)

  基站将该字段设置为 00000000。

6. 基站查询确认指令

基站查询确认指令可在寻呼信道或前向业务信道上发送。

基站用下列定长格式表示具体指令字段：

- 指令码(6 比特)

  基站将该字段设置为 000010。
- 指令限定码(8 比特)

  基站将该字段设置为 00000000。
- 查询响应(18 比特)
- 保留比特(6 比特)

  基站将该字段设置为 000000。

7. 数据突发消息

基站在寻呼信道上发送数据突发消息，要采用下面的变长消息格式：

字　段	长度(比特)
消息类型(00001010)	8
确认序列号	3
消息序列号	3
要求确认标识符	1
有效确认标识符	1
地址类型	3
地址字段长度	4
移动台地址	8 × 地址字段长度
消息号	8
数据突发类型	6
数据突发消息流的消息数	8
消息字符数	8
字符	8
保留比特(00000)	5

8. 切换导引消息

基站发送切换导引消息，要采用下面的变长消息格式：

字　段	长度(比特)
消息类型(00000101)	8
确认序列号	3
消息序列号	3
要求确认标识符	1
消息加密标识符	2
使用激活时间标识符	1
激活时间	6
切换导引消息序列号	2
激活集与候选集搜索窗大小	4
导频检测门限	6
导频撤销门限	6
激活集对候选集的比较门限	4
撤销时间值	4
帧偏移	4
专用长码掩码标识符	1
复位确认过程要求	1
消息加密模式	2
备用频段标识符	1
CDMA 信道频段	0 或 11
保留比特	0～7(依需要而定)

■下面记录的是一个或更多的具体值：

字　段	长度(比特)
导频 PN 序列偏移指数	9
功率控制码元组合标识符	1
编码信道索引	8

9. 业务状态系统参数消息

基站发送业务状态系统参数消息，要采用下面的定长消息格式：

字　段	长度(比特)
消息类型(00000111)	8
确认序列号	3
消息序列号	3
要求确认标识符	1
消息加密标识符	2
系统识别号(SID)	15
网络识别号(NID)	16
激活集与候选集的搜索窗大小	4
邻近集搜索窗大小	4
导频检测门限	4
导频撤销门限	6
激活集对候选集的比较门限	6
撤销计时器值	4
邻近集项最大保留寿命	4
保留比特	4
	4

10. 本地控制指令

本地控制指令可在寻呼信道或前向业务信道上发送。基站用下列定长格式表示具体指令字段：

- 指令码(6 比特)

  基站将该字段设置为 011110。

- 指令限定码(8 比特)

  具体的指令由每个系统确定的 nnnnnnnn 给出。

11. 锁定指令

锁定指令可在寻呼信道或前向业务信道上发送。基站用下列定长格式表示具体指令字段：

- 指令码(6 比特)

  基站将该字段设置为 010010。
- 指令限定码(8 比特)

  基站将该字段设置为 0001nnnn,其中 nnnn 为锁定原因。

12. 长码转移请求指令

长码转移请求指令只能在前向业务信道上发送。

- 指令码(6 比特)

  基站将该字段设置为 010111。
- 指令限定码(8 比特)

  基站将该字段设置为 00000000。

13. 维护指令

基站进入等待应答了状态。

14. 要求维护指令

要求维护指令可在寻呼信道或前向业务信道上发送。基站用下列定长格式表示具体指令字段:

- 指令码(6 比特)

  基站将该字段设置为 010010。
- 指令限定码(8 比特)

  基站将该字段设置为 0010nnnn,其中 nnnn 为维护原因。

15. 消息加密模式指令

消息加密模式指令只能在前向业务信道上发送。

- 指令码(6 比特)

  基站将该字段设置为 000011。
- 指令限定码(8 位)

  基站将该字段设置为 000000nn,其中 nn 为由下表得到的模式值。

加密模式字段(二进制)	采用的加密模式
00	不加密
01	加密呼叫控制消息

基站将"指令限定码"字段设置为上表所示的"加密模式"值,对应于用于前向和反向业务信道消息的加密模式。

16. 移动台注册消息
17. 相邻列表更新指令
18. 参数更新指令

参数更新指令只能在前向业务信道上发送。基站用下列定长格式表示具体指令字段:

- 指令码(6 比特)

  基站将该字段设置为 000101。
- 指令限定码(8 比特)

基站将该字段设置为 0000nnnn，其中 nnnn 为请求号。

19. 导频测量请求指令

导频测量请求指令只能在前向业务信道上发送。基站用下列定长格式表示具体指令字段：

- 指令码（6 比特）

  基站将该字段设置为 010001。

- 指令限定码（8 比特）

  基站将该字段设置为 00000000。

20. 功率控制参数消息

21. 释放指令

释放指令可在寻呼信道或前向业务信道上发送。基站用下列定长格式表示具体指令字段：

- 指令码（6 比特）

  基站将该字段设置为 010101。

- 指令限定码（8 比特）

  该字段为 00000000 表示没有给定的原因。

  该字段为 00000010 表明申请的服务选择被拒绝。

之后基站进入释放子状态。

22. 检索参数消息

23. 服务选择控制指令

服务选择控制指令只能在前向业务信道上发送。基站用下列定长格式表示具体指令字段：

- 指令码（6 比特）

  基站将该字段设置为 011101。

- 指令限定码（8 比特）

  基站将该字段设置为 nnnnnnnn。

  具体的控制由每个服务选择确定的 nnnnnnnn 给出。

24. 服务选择申请指令

服务选择申请指令只能在前向业务信道上发送。基站用下列定长格式表示具体指令字段：

- 指令码（6 比特）

  基站将该字段设置为 010011。

- 指令限定码（8 比特）

  基站将该字段设置为 00000000。

25. 服务选择响应指令

服务选择响应指令只能在前向业务信道上发送。基站用下列定长格式表示具体指令字段：

- 指令码(6 比特)

  基站将该字段设置为 010100。
- 指令限定码(8 比特)

  基站将该字段设置为 00000000。
- 服务选择

  基站将该字段设置为 TSB 58 中的服务选择码,对应于被接受的服务选择,或设置为 0000000000000000,以拒绝移动台申请的最近一个服务选择。

26. 设置参数消息
27. SSD 更新消息

基站在寻呼信道上发送 SSD 消息,要用下面的定长消息格式:

字　段	长度(比特)
消息类型(00001011)	8
确认序列号	3
消息序列号	3
要求确认标识符	1
有效确认标识符	1
地址类型	3
地址字段长度	4
移动台地址	8 × 地址字段长度
用于计算 SSD 的随机数据	56
保留比特	3

28. 状态请求指令

状态请求指令只能在前向业务信道上发送。状态请求指令的"指令限定码"字段指定了状态消息中移动台将返回的信息记录。基站用下列变长格式表示具体指令字段:

具体指令字段	长度(比特)
指令限定码	8
系统识别号(SID)	0 或 15
网络识别号(NID)	0 或 16
保留比特	0 或 1(依需要而定)

- 指令限定码(8 比特)

  基站将该字段设置为下表中的指令限定码,对应于状态消息中移动台将返回的信息记录类型。

所需信息记录	指令限定码(二进制)
识别号	00000110
呼叫模式	00000111
终止信息	00001000
MIN 信息	00001001
安全状态	00001010

- 系统识别号(SID)

  如果"指令限定码"字段设置为 00000110,基站就将该字段设置为该蜂窝系统的系统识别号。否则基站就忽略该字段。
- 网络识别号(NID)

  该字段用做由系统标识所有者定义的系统子标识符。如果指令限定码字段设置为

00000110，基站就将该字段设置为该网络的网络识别号。值为 65535 的 NID 作为预留。否则基站就忽略该字段。

- 保留比特

  基站按需要加入保留比特，以使具体指令字段的长度等于八位的整数。基站将这些保留比特设置为 0。

如果基站收到下列来自移动台的自动消息中的一个，基站就要按照具体要求对该消息进行处理：

1. 数据突发消息
2. 切换完成消息

   基站在（从激活集中移出的）前向业务信道上继续向移动台发射信号，直到基站收到移动台发来的切换完成消息，或者基站确定呼叫已经被释放。基站收到切换完成消息之后，才能放弃在（从激活集中移出的）前向业务信道上继续向移动台发射信号。
3. 长码转移请求指令

   如果发起消息或寻呼响应消息中申请了话音保密，基站可发送长码转移请求指令，以请求专用长码转移。
4. 参数更新证实指令
5. 导频强度测量消息

   基站可在导频强度测量消息中用导频强度测量来确定新的激活集。

   基站可通过发送切换导引消息来响应来自移动台的导频强度测量消息。
6. 功率测量报告消息

   如果基站启用了前向业务信道功率控制，移动台就用功率测量报告消息向基站报告帧差错率统计数字。
7. 释放指令

   基站在 T_{2b} = 0.8 秒之内向移动台发送释放指令，并进入释放子状态，或者在 T_{2b} 秒之内发送信息示警消息，并进入等待应答子状态。T_{2b} 表示基站收到一个释放指令后发送释放指令的最长时间。
8. 请求模拟服务指令

   基站可用模拟切换导引消息来进行响应。
9. 服务选择控制指令

   如果有一个服务选择是处于激活状态的，基站就会按照该激活服务选择的要求来处理收到的服务选择控制指令。
10. 服务选择申请指令

    基站处理服务选择申请指令，要进行下面的过程。如果基站没有接受移动台所申请的服务选择，并且有一个可申请的待选服务选择，基站就会将服务选择申请数设置为该待选服务选择数，并在 T_{4b} = 5 秒内发送一个服务选择申请指令以申请待选服务选择。T_{4b} 表示基站对服务选择申请的最长响应时间。
11. 服务选择响应指令

    如果收到服务选择响应指令，基站会执行后面的操作。如果该指令表明服务选择申请被

拒绝，基站就会将服务选择申请数设置为 NULL。接着基站和先前一样继续处理主业务，而后才会再接受服务选择申请指令，同时基站仍然处于目前的状态。

6.7　等待应答子状态

在此子状态，基站等待来自移动台的接通指令。在等待应答子状态，基站要进行下列操作：

- 基站发射功率控制子状态。
- 如果有一个服务选择处于激活状态（服务选择当前值不等于 NULL），基站就按照激活服务选择的要求来处理收到的主业务比特。否则，基站就丢弃收到的主业务比特。
- 如果有一个服务选择处于激活状态（服务选择当前值不等于 NULL），基站就按照激活服务选择的要求发射主业务比特。否则，基站就发射零业务信道数据。
- 基站进行消息确认过程。
- 如果基站声明反向业务信道连续性出现损耗，它就要向移动台发送释放指令。如果基站发送了释放指令，则基站进入释放子状态。
- 基站可进行前向业务信道功率控制。
- 基站可申请服务选择。为此，基站要进行服务选择申请初始化，以指明申请的服务选择号，并发送服务选择申请指令，即所申请的服务选择号。
- 如果有一个服务选择处于激活状态（当前服务选择号不等于 NULL），基站可发送服务选择控制指令（指令限定码），以按照激活服务选择的要求调用服务选择特殊功能。
- 基站请求长码转移，可以是自动的，也可以是为了响应（发起消息或寻呼响应消息中所指明的）话音保密请求。
- 基站可进行鉴权过程。
- 基站可发送下列消息，这些消息在 6.6 节中已探讨过。基站发送消息，要遵循以下具体要求：

1. 信息示警消息
2. 模拟切换导引消息
3. 监察指令
4. 鉴权查询消息
5. 基站确认指令
6. 基站查询确认指令
7. 数据突发消息
8. 切换导引消息
9. 业务状态系统参数消息
10. 本地控制指令
11. 锁定指令
12. 长码转移请求指令
13. 维护指令
14. 要求维护指令

15. 消息加密模式指令
16. 移动台注册消息
17. 相邻列表更新消息
18. 参数更新指令
19. 导频测量请求指令
20. 功率控制参数消息
21. 释放指令
22. 检索参数消息
23. 服务选择控制指令
24. 服务选择申请指令
25. 服务选择响应指令
26. 设置参数消息
27. SSD 更新消息
28. 状态请求指令

- 如果基站收到下列来自移动台的自动消息中的一个，基站就要按照具体要求对该消息进行处理：

1. 接通指令

   基站进入会话子状态。
2. 数据突发消息
3. 切换完成消息

   基站在（从激活集中移出的）前向业务信道上继续向移动台发射信号，直到基站收到移动台发来的切换完成消息，或者基站确定呼叫已经被释放。

   基站收到切换完成消息之后，才能放弃在（从激活集中移出的）前向业务信道上继续向移动台发射信号。
4. 长码转移请求指令

   如果发起消息或寻呼响应消息中申请了话音保密，基站可发送长码转移请求指令，以请求专用长码转移。
5. 参数更新证实指令

   参数更新证实指令只能在前向业务信道上发送。基站用下列定长格式码：

   - 指令码（6 比特）

     基站将该字段设置为 000101。
   - 指令限定码（8 比特）

     基站将该字段设置为 0000nnnn，其中 nnnn 为请求号。

6. 导频强度测量消息

   基站在导频强度测量消息中用导频强度测量来确定新的激活集。基站可通过发送切换导引消息来响应来自移动台的导频强度测量消息。
7. 功率测量报告消息

   如果基站启用了前向业务信道功率控制，移动台就用功率测量报告消息向基站报告帧

差错率统计数字。

8. 释放指令

基站在 T_{2b} 秒之内向移动台发送释放指令，并进入释放子状态，或者在 T_{2b} 秒之内发送信息示警消息，并进入等待应答子状态。

9. 请求模拟服务指令

基站可用模拟切换导引消息来进行响应。

10. 服务选择控制指令

如果有一个服务选择是处于激活状态的，基站就会按照该激活服务选择的要求来处理收到的服务选择控制指令。

11. 服务选择申请指令

如果基站没有接受移动台所申请的服务选择，并且有一个可申请的待选服务选择，基站就会将服务选择申请数设置为该待选服务选择数，并在 T_{4b} 秒内发送一个服务选择申请指令以申请待选服务选择。

12. 服务选择响应指令

如果收到服务选择响应指令，基站就执行下面的操作：

如果该指令表明服务选择申请被拒绝，基站就会将服务选择申请数设置为 NULL。接着基站和先前一样继续处理主业务，而后才会再接受服务选择申请指令，同时基站仍然处于目前的状态。

6.8　会话子状态

在此子状态，基站与移动台始呼服务选择请求交换主业务比特。进入会话子状态时，基站进行下列操作：

- 如果呼叫是由移动台发起的，则基站处理发起消息指定的服务选择申请。

处于会话子状态时，基站进行下列操作：

- 基站发射功率控制子信道。
- 如果有一个服务选择处于激活状态（服务选择当前值不等于 NULL），基站就按照激活服务选择的要求来处理收到的主业务比特。否则，基站就丢弃收到的主业务比特。
- 如果有一个服务选择处于激活状态（服务选择当前值不等于 NULL），基站就按照激活服务选择的要求发射主业务比特。否则，基站就发射零业务信道数据。
- 基站进行消息确认过程。
- 如果基站声明反向业务信道连续性出现损耗，它就要向移动台发送释放指令。如果基站发送了释放指令，则基站进入释放子状态。
- 基站可进行前向业务信道功率控制。
- 基站可申请服务选择。为此，基站要进行服务选择申请初始化，以指明申请的服务选择号，并发送服务选择申请指令（所申请的服务选择号）。
- 如果有一个服务选择处于激活状态（当前服务选择号不等于 NULL），基站可发送服务选择控制指令（指令限定码），以按照激活服务选择的要求调用服务选择特殊功能。

- 基站请求长码转移，可以是自动的，也可以是为了响应（发起消息或寻呼响应消息中所指明的）话音保密请求。
- 基站可进行鉴权过程。
- 基站可发送下列消息。基站发送消息，要遵循以下具体要求：

1. 信息示警消息
   如果消息包含一个信号类型字段为 01 或 10 的信号信息记录，或者消息不包含信号信息记录，则基站进入等待应答子状态。
2. ■模拟切换导引消息
   之后基站开始对话任务。
3. ■监察指令
4. ■鉴权查询消息
5. ■基站确认指令
6. ■基站查询确认指令
7. ■连续 DTMF 双音频指令
8. ■数据突发消息
9. ■信息闪动消息
10. 切换导引消息
11. 业务状态系统参数消息
12. 本地控制指令
13. 锁定指令
14. 长码转移请求指令
15. 维护指令
    基站进入等待应答子状态。
16. 要求维护指令
17. 消息加密模式指令
18. 移动台注册消息
19. 相邻列表更新消息
20. 参数更新指令
21. 导频测量请求指令
22. 功率控制参数消息
23. 释放指令
    基站进入释放子状态。
24. 检索参数消息
25. 发送突发 DTMF 消息
26. 服务选择控制指令
27. 服务选择申请指令
28. 服务选择响应指令
29. 设置参数消息
30. SSD 更新消息

31. 状态请求指令

- 如果基站收到下列来自移动台的自动消息中的一个，则基站按照具体要求对该消息进行处理：

1. ■连续 DTMF 双音频指令
2. ■数据突发消息
3. ■信息闪动消息
4. ■切换完成消息
5. ■长码转移请求指令
6. ■始呼继续消息
7. ■参数更新证实指令
8. ■导频强度测量消息
9. ■功率测量报告消息
10. 释放指令
    基站在 T_{2b} 秒之内向移动台发送释放指令，并进入释放子状态，或者在 T_{2b} 秒之内发送信息示警消息，并进入等待应答子状态。
11. 请求模拟服务指令
    基站可用模拟切换导引消息来进行响应。
12. 发送突发 DTMF 消息
13. 服务选择控制指令
14. 服务选择申请指令
15. 服务选择响应指令

6.9 释放子状态

在此子状态，基站切断呼叫。处于释放子状态时，基站进行下列操作：

- 基站发射功率控制子信道。
- 基站至少用 T_{3b} = 0.3 秒发射零业务信道数据。此时间间隔之后，基站将停止在前向业务信道上发射信号。T_{3b} 表示基站发送或接收释放指令后在编码信道上继续发射信号的最短时间。
- 基站进行消息确认过程。
- 基站可进行前向业务信道功率控制。
- 基站可发送下列消息。基站发送消息，要遵循以下具体要求：

1. ■信息示警消息
    如果消息包含一个信号类型字段为 01 或 10 的信号信息记录，或者消息不包含信号信息记录，则基站进入等待应答子状态。
2. ■监察指令
3. ■基站确认指令
4. ■数据突发消息
5. ■切换导引消息
6. ■业务状态系统参数消息

7. ■本地控制指令
8. ■锁定指令
9. ■维护指令
10. 要求维护指令
11. 移动台注册消息
12. 相邻列表更新消息
13. 参数更新消息
14. 功率控制参数消息
15. 释放指令
16. 检索参数消息
17. 服务选择控制指令
18. 状态请求指令

- 如果基站收到下列来自移动台的自动消息中的一个，则基站按照具体要求对该消息进行处理：

1. ■接通指令
2. ■连续 DTMF 双音频指令
   双音频（DTMF）指明了通过同时发射两个音频来发信号的方法，其中一个音频来自低频组，另一个来自高频组。每个频率组都由四个频率组成。
3. ■数据突发消息
4. ■信息闪动消息
5. ■切换完成消息
6. ■导频强度测量消息
7. ■功率测量报告消息
8. ■长码转移请求指令
9. ■始呼继续消息
10. 释放指令
11. 请求模拟服务指令
12. 发送突发 DTMF 消息
13. 服务选择控制指令
14. 服务选择申请指令
15. 服务选择响应指令

6.10 注册

注册就是移动台向基站告知其位置、状态、识别号、时隙周期以及其他特性的过程。基站建立以移动台为终点的呼叫时，用位置信息来对移动台进行有效率的寻呼。同时注册还为移动台提供时隙周期指数参数，使基站能够确定时隙化模式下移动台监视的是哪些寻呼信道时隙。注册还提供移动台等级标记和协议修改号，使基站得知移动台的容量。

CDMA 系统支持 9 种不同形式的注册。

1. 开电源注册
   移动台开电源，退出备用服务系统或模拟系统，都要进行注册。
2. 关电源注册
   如果先前移动台在当前的服务系统中进行过注册，则关电源时也要注册。
3. 基于计时器的注册
   如果计时器达到规定时间，移动台要进行注册。
4. 基于距离的注册
   如果当前基站和上一次注册的基站之间的距离超过了门限值，移动台要进行注册。
5. 基于区域的注册
   移动台进入一个新的区域要进行注册。
6. 参数改变注册
   如果移动台存储的某些参数发生了改变，要进行注册。
7. 按指令注册
   如果基站请求移动台进行注册，移动台要进行注册。
8. 默认注册
   如果移动台成功地向基站发送了发起消息或寻呼响应消息，基站就能推断出移动台的位置，这就叫做“默认注册”。
9. 业务信道注册
   如果基站有移动台的注册信息，而该移动台已被指配给一个业务信道，基站可告知该移动台“已被注册”。

   前五种形式的注册作为一组，称做“自主注册”，部分决定于漫游状态和系统参数消息中包含的标识符。基站通过指令消息来启动按指令注册。

   如果移动台已被指配给一个业务信道，则基站可通过状态请求指令获得移动台的注册信息，从而得到移动台的状态消息。基站可通过移动台注册的消息来通知移动台“已被注册”。

6.10.1　寻呼和接入信道上的注册

基站在系统参数消息中规定启用的注册形式、相应的注册参数和漫游状态情况。如果启用了任何一种自主注册形式，基站也应启动基于参数的注册。

基站将接入信道上发送的发起消息或寻呼响应消息作为发送消息的移动台的默认注册来处理。通过向移动台发送注册请求指令，基站能随时获得移动台的全部注册信息。

6.10.2　业务信道上的注册

基站通过状态请求指令获得移动台在业务信道上的注册信息。如果基站已有移动台的注册资料，基站可向移动台发送一个“移动台已注册”消息，指明基站的注册系统、区域和位置信息。

6.11 切换过程

这一节讲述在切换过程对移动台的要求。基站支持下列三种切换过程：

- 软切换

  在这种切换中，当移动台开始与一个新的基站通信时，并不中断与原来基站之间的通信。只有当所有移动台指配前向信道具有相同频率时，基站才能命令移动台进行软切换。软切换可提供在基站边界处的前向业务信道和反向业务信道的路径分集。

- CDMA 到 CDMA 的硬切换

  在这种切换中，基站命令移动台在不相交的基站集合之间、不同的频率之间或不同帧偏移之间进行转换。

- CDMA 到模拟系统的硬切换

  在这种切换中，移动台命令基站从前向业务信道上转换到模拟话音信道上。

激活集包含了与移动台指配前向业务信道相对应的导频。基站把使用信道指配消息和切换导引消息的激活集内容告知移动台。

6.11.1 开销消息

基站发送下列消息，以管理移动台所进行的导频搜索过程：

- 系统参数消息

  基站在寻呼信道上的系统参数消息中发送切换相关参数。

- 业务状态系统参数消息

  基站可修改工作在业务信道上的移动台的切换相关参数，通过发送业务状态系统参数消息来实现。

- 相邻列表消息

  基站在寻呼信道上的相邻列表消息中发送相邻列表。基站必须在相邻列表消息中列出超过 N_{8m} = 20 个的导频。

- 相邻列表更新消息

  基站可修改工作在业务信道上的移动台的相邻列表，通过发送相邻列表更新消息来实现。基站发送的相邻列表更新消息中不包括移动台激活集中的导频。基站在相邻列表更新消息中列出的导频不能多于 N_{8m} 个。基站在相邻列表更新消息中将导频按优先级降序列出。对邻近集中的每一个导频，移动台都给出一个计数器和相邻列表寿命。当移动台将导频从激活集或候选集移到邻近集时，要将计数器初始化为零。当移动台将导频从剩余集移到邻近集时，要将计数器初始化为邻近集最大寿命。移动台收到相邻列表消息后，会将邻近集中的每个导频的相邻列表寿命加 1。

- 切换导引消息

  基站也可以通过切换导引消息修改下列参数的值：

  1. 激活集和候选集的搜索窗大小

     基站将该字段设置为“搜索窗大小”参数，对应于移动台为进行激活集和候选集的导频搜

索而达到的 PN 码片数。

2. 导频检测门限

移动台用该值来触发导频强度测量消息的发送，以启动切换过程。

3. 导频撤销门限

移动台用该值来触发导频强度测量消息的发送，以发射切换过程，并使更多的导频从候选集移到邻近集。

4. 激活集对候选集的比较门限

当候选集中的导频强度超过激活集中的导频强度，移动台会根据二者差值发射导频强度测量消息。基站将该字段设置为门限候选集导频和门限激活集导频的比值。

5. 撤销计时器值

计时器达到该值后，移动台会对导频（激活集或候选集中的项，其强度不大于撤销计时器的值）采取操作。如果该导频是激活集中的项，移动台就发布导频强度测量消息。如果该导频是候选集中的项，移动台就将其移入邻近集。

6.11.2　切换过程中的呼叫处理

1. 导频强度测量消息的处理

基站可在导频强度测量消息中用导频强度测量来确定新的激活集。

基站可在导频强度测量消息中用 PN 相位测量来估计对移动台的传播时延。这种估计可减少 RTC 捕获时间。

基站可通过发送切换导引消息来响应来自移动台的导频强度测量消息。

2. 切换导引消息的处理

基站保留了一个切换导引消息序列号。向移动台发射第一个切换导引消息之前，应将该序列号初始化为零。基站每次修改发给移动台的切换导引消息中的导频列表，都要将该序列号加 1（模 4）。

一次硬切换之后，基站应将切换导引消息（HDM）序列号设置为上一个切换完成消息的 HDM 字段，并用切换完成消息中包含的导频指令来解释随后的功率测量报告消息的内容。

基站按照下列规则设置切换导引消息的内容：

- 一个切换导引消息在新的激活集中列出的导频不超过 $N_{6m}=6$ 个。N_{6m} 代表基站支持的业务信道激活集大小。
- 切换导引消息可识别相同的功率控制子信道。
- 如果 CDMA 频段是固定不变的，切换导引消息不会改变留在新激活集里的激活集导频所对应的编码信道。
- 基站用切换导引消息的专用长码掩码字段来指定在新的前向业务信道中将使用的长码掩码。基站可改变该字段的内容（仅对于 CDMA 到 CDMA 的硬切换）。如果基站已在切换导引消息中指定了长码掩码，但没有指定显性作用时间，基站将在第一个 80 ms 的时间界限内（相对于系统时间）使用新的长码掩码（此过程在包含消息最后一个比特的帧结束后至少 80 ms 发生）。
- 对于 CDMA 到 CDMA 的硬切换，基站可通过使切换导引消息的第二层字段复位来要求移动台执行确认过程复位。如果基站要求移动台执行确认过程复位，它自己也应将确认过

程复位。

- 对于 CDMA 到 CDMA 的硬切换，基站可通过将帧偏移字段设置为新值来改变帧偏移。如果基站规定了新的帧偏移，但没有规定显性作用时间，基站要在发射完切换导引消息后的第二个 80 ms 的时间界限内(相对于系统时间)改变前向业务信道和反向业务信道帧偏移，除非切换导引消息刚好是在 80 ms 的时候发射完，在这种情况下，则应在发射完切换导引消息后 80 ms 时改变帧偏移。

3. 切换过程的信号发射

   基站会不断在移出激活集的前向业务信道上向移动台发射信号，直到收到移动台的切换完成消息，或确定呼叫已经释放。基站收到移动台的切换完成消息后，就会停止在移出激活集的前向业务信道上向移动台发射信号。

4. 令移动台发送导频强度测量消息

   基站可通过发送导频测量请求指令来命令移动台发送导频强度测量消息。

6.11.3 激活集维护

基站对其控制的各移动台的激活集都要进行如下维护：

- 基站发送信道指配消息时，要将激活集初始化为只包含指配前向业务信道对应的导频。
- 基站发送切换导引消息时，要在消息作用时间之前将消息中所有命名了的导频加入激活集，如果它们还不在激活集里的话。
- 收到切换完成消息后，基站会从激活集中删除那些在最近的切换导引消息中没有命名的导频。

6.11.4 软切换

基站命令移动台从一个前向业务信道向相同频段的另一个前向业务信道转换时，进行软切换。

1. 软切换过程的信号接收

   激活集中的各基站都进行反向业务信道解调。基站应提供分集，以合并从激活集中的各基站得到的解调信号。

2. 软切换过程的信号发射

   通过消息作用时间，基站开始在切换导引消息中指定的所有前向业务信道上发射相同的调制码元(功率控制子信道有可能例外)。

基站在所有相同的功率控制子信道上发射相同的功率控制码元，这些子信道的识别与上一个切换导引消息相同。

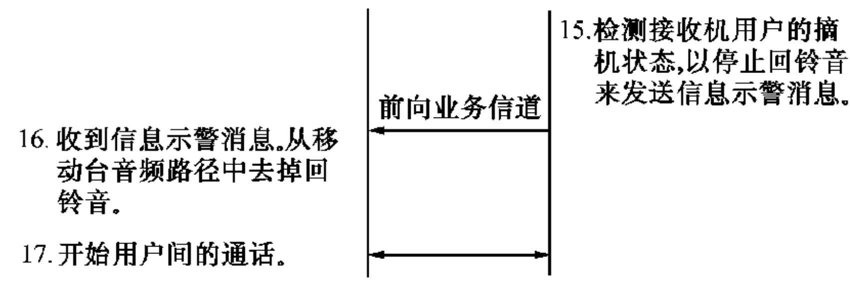
在激活集导频对应的反向业务信道和所有前向业务信道上，基站使用相同的长码掩码。

6.11.5 CDMA 到模拟系统的硬切换

基站可通过发送模拟切换导引消息来命令移动台进行 CDMA 系统到模拟系统的转换。

例 6.1　下面的简化呼叫流程展示了移动台发起呼叫的例子

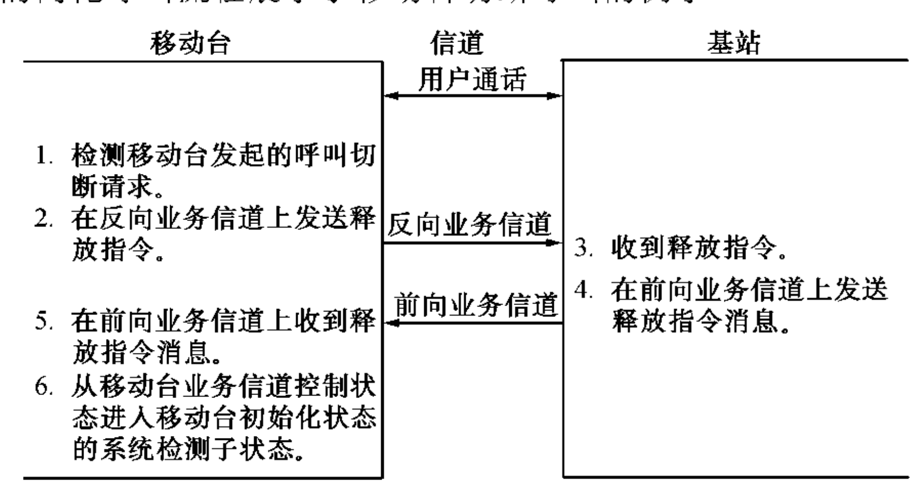
移动台	信道	基站
1. 检测用户发起的呼叫		
2. 在接入信道上发送发起消息。其中包括移动台的 ESN 和 MIN，被叫方号码的拨号数字、特殊服务 (0)、发起服务选择(16 个 0 的序列)、鉴权数据 (Authr)。	接入信道 →	3. 使用 Authr 进行鉴权验证过程，然后建立业务信道。
	前向业务信道 ←	4. 开始发送由 16 个 1 和 8 个 0 的序列组成的零业务信道数据。
		5. 在寻呼信道上发送信道指配消息。其中包括 ESN、编码信道、CDMA 频段以及帧偏移。
6. 收到信道指配消息。捕获 CDMA 频段、编码信道，以及帧偏移。建立业务信道。使用公共长码在前向业务信道上收到 N_{5m}(=2) 个连续的有效帧偏移，开始发送业务信道报头。	寻呼信道 ←	
	反向业务信道 →	7. 捕获反向业务信道上的业务信道报头。发送基站确认指令。
8. 收到基站确认指令。开始发射零业务信道数据。	前向业务信道 ←	
	反向业务信道 →	9. 收到零业务信道数据。在前向业务信道上发送服务选择响应指令。其中包含的服务代号为 1。
10. 收到服务选择响应指令。开始根据服务选择代号(1)处理主业务比特。	前向业务信道 ←	
11. 以一个拨号数字来发送发起继续消息。	反向业务信道 →	12. 收到发起继续消息。
		13. 用回铃音发送信息示警消息，以便将回铃音应用于移动台的音频路径。
14. 收到信息示警消息。将回铃音应用于音频路径中。	前向业务信道 ←	



例 6.2　下面的简化呼叫流程展示了移动台为呼叫终点的例子

移动台	信道	基站
		1. 检测用户终止呼叫。在寻呼信道上发送寻呼消息或时隙化寻呼消息，其中包含 MIN1,MIN2 以及特殊服务。
	← 寻呼信道	
2. 收到寻呼消息或时隙化寻呼消息。在接入信道上发送寻呼响应消息。其中包括 MIN、ESN、鉴权数据 (Authr)，以及服务选择。		
	接入信道 →	3. 用 Authr 和 Ranc 进行鉴权验证过程建立业务信道。
		4. 开始发送由 16 个 1 和 8 个 0 的序列组成的零业务数据。
	← 前向业务信道	
	← 寻呼信道	5. 在寻呼信道上发送信道指配消息。其中包含 ESN、CDMA 编码信道，以及帧偏移。
6. 收到信道指配消息。捕获CDMA编码信道和帧偏移。用公共长码建立业务信道。在前向业务信道上收到 $N_{5m}(=2)$ 个连续的有效帧。开始发送业务信道报头。		
	反向业务信道 →	7. 在业务信道上捕获业务信道报头。发送基站确认指令。
	← 前向业务信道	
8. 收到基站确认指令。开始发射零业务信道数据。		
	反向业务信道 →	9. 收到零业务信道数据。在前向业务信道上发送服务选择响应指令。其中包含的服务选择代号为 1。
	← 前向业务信道	
10. 收到服务选择响应指令。开始根据服务选择代号 (1) 处理主业务比特。		
		11. 用振铃来发送信息示警消息，以便将振铃应用于移动台。
	← 前向业务信道	
12. 收到信息示警消息。开始向基站送出振铃。检测用户摘机状态，即回答呼叫。停止振铃。在反向业务信道上发送接通指令。		
	反向业务信道 →	13. 收到接通指令。
14. 开始用户间的通话。	←→	

例 6.3　下面的简化呼叫流程展示了移动台切断呼叫的例子



第 7 章　单向 Hash 函数和消息摘要简述

本章介绍一种 Hash 函数。这种 Hash 函数适用于安全技术，尤其适用于提供鉴权和数字签名。

Hash 函数是一种单向函数。单向 Hash 函数也称做消息摘要函数。因此，可用术语 Hash 码和消息摘要来作为替代的名称。Hash 函数以一个大小可变的消息 M 作为输入，并将其变换为固定大小的消息摘要 $H(M)$，作为输出。通常 $H(M)$ 要比 M 小很多。例如，$H(M)$ 可以是 18，64 或 128 比特，而 M 却可以为兆字节或更多。

对消息鉴权来说，消息摘要函数常常是相当重要的，因此理解下面所叙述的性质也是很有用的：

1. ■Hash 函数 H 适用于变长消息 M 的任何数据块。
2. ■Hash 函数 H 产生一个定长的 Hash 码 $H(M)$。
3. ■$H(M)$ 应按用户特殊保密要求给定的 M 来计算。
4. ■要找到有预先指定消息摘要 $H(M)$ 的消息 M，在计算上应当是不可行的。
5. ■对于任意给定的消息 M，要找到另一个消息 M'，使 $H(M) = H(M')$，在计算上也应当是不可行的。换句话说，要找到映射到同一个消息摘要 $H(M) = H(M')$ 的两个不同的 M 和 M'（M 不等于 M'），在计算上是不可行的。满足这个性质的 Hash 函数称为抗碰撞的 Hash 函数。

7.1　用于加密算法的单向函数

下面列出设计加密算法的一些方法：

- 离散对数问题

  给定 x，幂 g^x（模 p）较容易计算，但反过来要计算 x 就困难多了。
- 质数因式分解问题

  将两个大的质数相乘得到乘积是很容易的，但要将乘积因式分解得到两个大的质数就比较困难了。
- 陷门背包问题

  背包体制是由 Merkle 和 Hellman 提出的，已被建议用于公钥密码体制的设计。Merkle－Hellman 方法采用了超递增序列。

  超递增序列 $\{K_i\}(1 \leqslant i \leqslant n)$ 中每一项都比前面所有项的和大，即：

  $$K_i > K_1 + K_2 + \cdots + K_{i-1}, 1 \leqslant i \leqslant n$$

  给定一个二进制数 x_i（0 或 1），$1 \leqslant i \leqslant n$，且和 $Y = \{y_i\}$ 为

  $$Y = x_1 K_1 + x_2 K_2 + \cdots x_n K_n$$

  由一串等长的 x_i 与 K_i 的乘积之和得到 Y 较容易计算，但要由 Y 和 K_i 得出 x_i 就很难做到了。相应的背包问题就是要由和序列 $Y = \{y_i\}$ 得出二进制数 $\{x_i\}$。超递增背包的一个解法如

下：

如果 $y_n < K_n$，则令 $x_n = 0$ 且 $y_{n-1} = y_n$。

如果 $y_n > K_n$，则令 $x_n = 1$ 且 $y_{n-1} = y_n - K_n$。

利用算出的 y_{n-1} 值，用类似方法可得出 x_{n-1} 和 y_{n-2}。这种反推过程要持续到算出 $X = (x_1, x_2, \cdots, x_n)$ 时为止。

掩盖背包问题，比如乘法陷门背包和多交互背包，可证明是安全的，但遗憾的是，上述陷门背包和其大多数变形都已经被破译。然而，Merkle – Hellman 背包算法的理论思想还是有其历史价值的。

- 模平方根问题

  如果 n 是两个质数的乘积，则计算模 n 平方根的效率与对 n 进行因式分解的效率在计算上是等同的。用户 A（知道 n 的质因数）可以轻松地算出一个数的模 n 平方根，但对于其他的用户来说，这种计算已经被证明和求 n 的质因数一样困难。

- 积分剩余问题

  如果 m 是质数且 $a < m$，则 a 为一个模 m 的积分剩余，如果存在 x，使 $x^2 = a \pmod m$。

  如果 $x^2 = a \pmod m$ 和 $\gcd(a, m) = 1$ 有解，则 a 称做二次剩余。积分剩余问题就是给出 x 和 m，很容易计算 x^2（模 m），但如果给出 a 和 m，却很难由它们算出 x。

7.2 鉴权数据的消息摘要算法

本节的目的在于提供一种适用于安全技术的 Hash 函数变形。下面就详细说明这种适用于提供鉴权的 Hash 函数。

1. MD2 算法

   MD2 的输入为一个长为 8 位字节整数的消息，产生一个 128 位的消息摘要。

   消息被填充为添加在末尾的 16 字节校验和的倍数。消息每次处理 16 字节，以产生一个中间消息摘要。摘要的每个中间值取决于前一个中间值和处理中的 16 字节发起消息。MD2 的结构与 MD4 和 MD5 相似，但速度更慢，安全性也差一些。

2. MD4 算法

   MD4 是在 1990 年由 Rivest 设计的，是一个 32 位的以“字”（word）来定位的方案。MD2 要求消息为整数个字节，而 MD4 却能处理任意数量的比特。因此，MD4 在 32 位的机器上就比以字节定位的 MD2 算法要快。

   MD4 以任意长度的消息为输入，输出 128 位的消息摘要。采用这种方法，要产生有相同消息摘要的两个消息，或产生预先规定的目标消息摘要的任何消息，在计算上都是不可行的。

   MD4 算法倾向于数字签名的应用。其中，在公钥密码体制下，一个大文件要在以安全方式压缩之后，才能用一个密钥来进行加密。

3. MD5 算法

   MD4 消息摘要算法是一个单向 Hash 函数，产生一个 128 位的 Hash。MD4 算法提出后，少数研究者就对 MD4 的第一轮进行了批评。Rivest 随后对算法进行了改进，改进后的算法称做 MD5。但 MD5 比 MD4 稍慢。

   MD5 实际上是 MD4 的改进版，但对于 512 位的输入数据块（分为 16 个 32 位的子块），它仍

产生 128 位的 Hash 值。MD5 的输出为一组的 4 个数据块，这些数据块级联为一个 128 位的消息摘要。

4. 安全 Hash 算法（SHA）

如同 MD5，SHA 也是 MD4 的一个变形，由 NIST（美国国家标准与技术研究所）和 NSA（美国国家安全局）共同设计。FIPS 提出一个安全 Hash 标准（SHS），规定了安全 Hash 算法（SHA），以确保数字签名算法（DSA）的安全性。当然，SHA 可以应用到任何需要安全 Hash 函数的地方。

5. Snefru

Snefru 是由 Merkle 设计的一种单向 Hash 函数，将任意长度的消息 Hash 变换成 128 位或 256 位的值。消息被分成长度为 512/m 的小段。该算法将一个 512 位的值 Hash 变换成一个 m 位的值。对于这样设计的 Hash 函数，要计算一个输入的 Hash 是比较容易的，但要求出产生特定 Hash 值的输入，在计算上却是不可行的。Biham 和 Shamir 采用两种微分破译方法论证了双向 Snefru 的不安全性。

6. Kerberos

Kerberos 最早是在 1988 年由 MIT 开发的，作为一种能够限制授权用户接入和服务申请鉴权的鉴权服务。

Kerberos 只依赖于常规加密，而不用到公钥加密。

Kerberos 的版本 4 和版本 5 总体设计较为相似。但版本 4（1988 年）是应用最为广泛的版本，且只能用于 TCP/IP 网络。而版本 5（1994 年）校正了版本 4 的一些安全性上的不足，并被作为一个草拟的互联网标准（RFC 510）而发布。

Kerberos 由一个密钥分配中心（KDC）和一个子程序库来实现。

7. X.509

X.509[CCIT87]最早是在 1988 年发布，以公钥密码体制和数字签名为基础。数字签名技术要求使用 Hash 函数。X.509 的目录服务的广泛应用前景使其成为一个重要的标准。

X.509 标准没有讲述什么特殊的 Hash 算法，而是介绍了 RSA。当然，X.509 后来改为进行一些安全业务的寻址，修改后的建议标准也于 1993 年发布。

8. DM 算法

1985 年，Davies 和 Meyer 独立提出基于 DM 算法的 Hash 码算法。要进行 Hash 变换的消息首先被分为定长的串：$M_1, M_2, \cdots, M_t$。消息串 $M_i(1 \leqslant i \leqslant t)$用做密钥。前一消息串就用这个密钥进行加密，然后与它自身进行异或运算，其结果作为下一轮的输入。这样，最后消息长度就变为最后一个 M_i 的长度。令 $H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1}, 1 \leqslant i \leqslant t$。采用 DES 算法，首先将消息 M 划分为 56 位的 DES 密钥串。第一个串 M_1 即为 56 位的 DES 密钥。DES 算法的输入为一个初始向量 H_0。DES 算法输出的密文为 H_1，是 $E_{M_1}(H_0)$与 H_0 的异或，作为第二个消息串 Hash 变换的输入。下一个 56 位的消息串 M_2 成为新的密钥。H_1 为算法中输入的明文，H_2 为输出的密文。$H_2 = E_{M_2}(H_1) \oplus H_1$ 为第三个消息串的输入，以此类推。

9. 改进的 CBC 算法

Hash 码的计算有很多方法，改进的 CBC 方案就是其中一种。这种改进的 CBC 算法在总体设计思想上有点类似于上述 DM 方案。要进行 Hash 变换的填充消息被分为 t 个串，$M_1, M_2, \cdots, M_t$。其中每个 $M_i(1 \leqslant i \leqslant t)$都是 64 比特长。令 H_0 为一个初始向量（IV）。然后进行下

面的类 DES 多重加密步骤：

$H_1 = E_{H_0 \oplus M_1}(H_0) \oplus H_0$

$H_2 = E_{H_1 \oplus M_2}(H_1) \oplus H_1$

$H_i = E_{H_{i-1} \oplus M_i}(H_{i-1}) \oplus H_{i-1}$

其中 $H_i \oplus M_i$ 表示密钥。

消息摘要算法在设计公钥密码体制或私钥密码体制中更加有用。用来计算长消息的数字签名就不太实际了，因为速度慢，性能又差。消息能被压缩成长度更短的消息摘要。因此，一般是计算压缩消息摘要的数字签名，而不是整个长消息的数字签名。

第 8 章　鉴权、加密和识别

数据安全包含保密和鉴权两方面的问题，保密是指防止非法用户从信道中提取信息，因此，必须保证信息或数据的保密性和完整性，这样才能得到正确的信息。鉴权要解决的问题是防止错误数据插入到信道中或数据被篡改。

移动台必须与基站协同操作对移动台进行鉴权，鉴权过程是通过交换移动台和基站之间的信息达到确认移动台身份的目的，一个成功的鉴权需要移动台和基站处理一组完全相同的共享加密数据。例如，在 CDMA 鉴权协议中，移动台和基站互相都有匹配的加密认证符（即相同的共享加密数据），这实际上是由鉴权算法产生和发布的符号的短消息摘要。

8.1　移动台识别号

ITU-T 建议 E. 212 根据国际移动台标识（IMSI）定义了移动台识别号（MSIN 或 MIN），IMSI 的结构如图 8.1 所示。



图 8.1　国际移动台标识（IMSI）

IMSI 最多可以有 15 位数字，它由两个部分组成。第一部分是 3 位数字的移动国家码（MCC），其余的部分是国家移动台标识（NMSI），在这个标准里，IMSI 包含 3 位数字的移动国家码、2 位数字的移动网络码（MNC）和 10 位数字的移动台识别号（MSIN）。

如图 8.2 所示，移动台识别号（MSIN）是一个 34 位的二进制数，它来源于 10 位数电话号码簿里的电话号码，步骤见下面的说明。

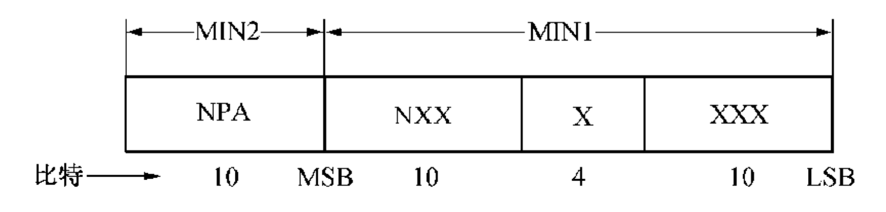


图 8.2　移动台识别号（MIN）

假设一个 10 位数的电话号码是 $D_1D_2D_3$-$D_4D_5D_6$-$D_7D_8D_9D_{10}$，其中 $D_1D_2D_3$ 是区域码，$D_4D_5D_6$ 是局域码，$D_7D_8D_9D_{10}$ 是个人电话号码。

1. 最先的 3 个数字通过下面的编码算法转换成 10 位二进制数（对应 MSIN2）：

- 用 3 位数字段表示 $D_1D_2D_3$，其中数字 0 的值为 10；
- 计算 $100D_1 + 10D_2 + D_3 - 111$；
- 用标准的十进制到二进制的换算将上面计算出的结果换算成二进制(见表 8.1)。

2. 接下去的 3 个数字用与(1)相同的方法转换成 10 位二进制数，作为 MSIN1 高位的 10 位数。
3. 最后 4 个数字通过下面的方法转换成 14 位二进制数，作为 MSIN1 低位的 14 位数：

- 第一位数字用二进制编码的十进制(BCD)方法直接转换成一个 4 位的二进制数，详细说明见表 8.1；
- 最后三位数字用与(1)相同的方法转换成 10 位二进制数。

表 8.1　十进制到二进制的转换和 **BCD** 映射

* 注：　MSIN 与 MIN 这两种表示方法通用。

十进制到二进制的转换		千位数字的 **BCD** 映射过程	
十进制数	二进制数	十进制数	二进制数
1	0000000001	1	0001
2	0000000010	2	0010
3	0000000011	3	0011
4	0000000100	4	0100
5	0000000101	5	0101
.	.	6	0110
.	.	7	0111
.	.	8	1000
998	1111100110	9	1001
999	1111100111	0	1010

例 8.1　假设用上述方法将一个10位数字的电话号码 290-453-7186编码成 MSIN2 和 MSIN1。

1. 计算 MSIN2：10 比特的 MSIN2 由电话号码的头 3 位数字(即 290)得到。

- $D_1 = 2, D_2 = 9, D_3 = 0$
- $100D_1 + 10D_2 + D_3 - 111 = 100(2) + 10(9) + 10 - 111 = 189$[①]
- 189 转换成二进制是 00 1011 1101。

因此，MSIN2 是 00 1011 1101。

2. 计算 MSIN1：MSIN1 的 10 个高位比特由电话号码的第二个 3 位数字(即 453)得到。

- $D_1 = 4, D_2 = 5, D_3 = 3$
- $453 - 111 = 342$
- 342 转换成二进制是 0101 0101 10。

3. 用 BCD 法转换 D_7：接下去的 MSIN1 的 4 个比特由电话号码千位上的数字(即 7)通过 BCD 转换得到：

- 7 的 BCD 转换码是 0111。

4. 计算 MSIN1 的最后 3 个数字(即 $D_8D_9D_{10}$)：MSIN1 低位的 10 个比特由电话号码的最后 3

① 注：此处原文为 100D1 + 10D2 + D3 - 111 = 100(2) + 10(9) + 0 - 111 = 179，179 转换成二进制是 00 1011 0011，因此，MSIN2 是 00 1011 0011，可能有误——译者注。

个数字(即 186)得到。

- $D_1 = 1, D_2 = 8, D_3 = 6$
- $100D_1 + 10D_2 + D_3 - 111 = 100(1) + 10(8) + 6 - 111 = 75$
- 75 转换成二进制是 00 0100 1011。

因此,MSIN1 是 0101 0101 1001 1100 0100 1011。

参考图 8.1,规定 MNC 的二进制转换如下:

1. ■D_1D_2 表示 2 位数字的移动网络码,若数字等于 0 则当做 10;
2. ■计算 $10D_1 + D_2 - 11$;
3. ■用标准的十进制到二进制的换算将步骤(2)中的结果换算成二进制。

接着,规定 MCC 的二进制转换:

1. ■$D_1D_2D_3$ 表示 3 位数字的移动国家码,若数字等于 0 则当做 10;
2. ■计算 $100D_1 + 10D_2 + D_3 - 111$;
3. ■用标准的十进制到二进制的换算将步骤(2)中的结果换算成二进制。

8.2 电子序列号(ESN)

ESN 是一个 32 位的二进制数,对任何蜂窝系统而言,它是移动台的惟一标识。ESN 的比特分配情况见图 8.3 所示。

ESN 必须由生产厂家设定,且其字段不易被修改,ESN 的修改需要一种专用设备,这种设备用户通常是得不到的。提供 ESN 的电路必须避免欺骗性的接触和篡改。将电子存储设备安装到槽里或是用一根电缆连接是不符合要求的,企图改变 ESN 电路必将破坏移动台的可操作性。

ESN 最初的类型认可发布时,32 位序列号最高位的 8 位(即位 31 到位 24)作为生产商代码分配给生产商,位 23 到位 18 保留(初始全 0),位 17 到位 0 由各个生产商进行惟一分配。当一家生产商已经充分地使用完从位 17 到位 0 范围内序列号的所有组合时,生产商将提交一份通知给美国政府的 FCC,FCC 将给他分配保留块(即位 23 到位 18)中的下一个连续二进制数。

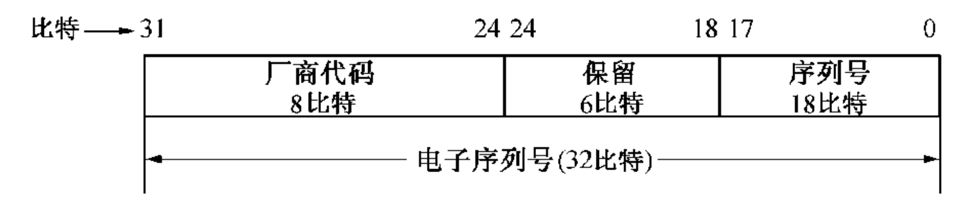


图 8.3 ESN 的比特分配

8.3 鉴权

鉴权过程是通过交换移动台和基站之间的信息达到确认移动台身份的目的。一个成功的鉴权需要移动台和基站处理一组完全相同的共享加密数据(SSD)。SSD 是存储于移动台(在半永久存储器中)中的 128 比特数据,且对基站而言,它是可用的。SSD-A 用于支持鉴权,SSD-B 用于支持话音和消息加密。

在这个标准里，定义了两个主要的鉴权过程：全局查询和惟一查询。

全局查询是在寻呼和接入信道上启动，移动台可以在接入信道上启动下列鉴权过程：

- 注册鉴权
- 发起呼叫鉴权
- 寻呼响应鉴权

另一方面，惟一查询将在前向和反向业务信道(W-CDMA 中的信令信道)上或者是寻呼和接入信道上启动，基站在下列情况下启动惟一查询响应过程：

- 注册鉴权失败
- 发起呼叫鉴权失败
- 寻呼响应鉴权失败
- 信道指配后的任何时候

表 8.2 总结了这个标准中对于每一种用法其鉴权算法的输入参数。

表 8.2　鉴权算法输入参数

过程	Rand-Challenge	ESN	AUTH-Data	SSD-AUTH	Save-registers
注册	RAND	ESN	MSIN1	SSD-A	False
惟一查询	256 × RANDU + (8LSBs of MSIN2)	ESN	MSIN1	SSD-A	False
发起呼叫	RAND	ESN	MSIN1	SSD-A	True
中断	RAND	ESN	MSIN1	SSD-A	True
基站查询	RANDBS	ESN	MSIN1	SSD-A-NEW	False

RAND：鉴权随机查询值(0 或 32 比特)

RANDU：惟一随机变量(24 比特)

RANDBS：随机查询数据(32 比特)

8.3.1　共享加密数据(SSD)

SSD 是存储于移动台半永久存储器中的 128 位数据，如图 8.4 所示，SSD 被分成两个不同的子集，每个部分用来支持不同的过程。

SSD-A 用于鉴权，SSD-B 用于话音保密和信息加密，SSD 的生成过程在后面的 8.12 节中有详细说明。

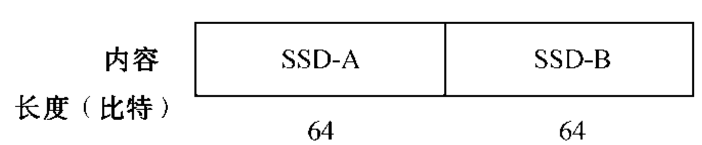


图 8.4　SSD 的划分

8.3.2　随机查询存储器(RAND)

RAND 是保存在移动台中的一个 32 位的值，当在 CDMA 寻呼信道上接收到它时，它等于收到的上一个接入参数消息的 RAND 值，接入参数消息定义了当移动台通过接入信道向基站发送消息时移动台所用到的参数。

RAND 与 SSD-A 以及其他适当的参数结合起来对移动台在发起呼叫、中断和注册时进行鉴权。

8.3.3 呼叫历史参数(COUNT)

呼叫历史参数定义了一个保存在基站中的模 64 的事件计数，当移动台在前向业务信道上接收到一个参数更新指令后，它将更新 COUNT。

8.4 移动台注册鉴权

当接入参数消息的鉴权字段值设为 01，并且移动台尝试在接入信道上发送注册请求消息进行注册时，与鉴权相关的过程将会执行如下：

移动台：

- 设置鉴权算法的输入参数，如图 8.5 所示；
- 设置存储注册输入参数为 FALSE；
- 执行鉴权算法；
- 设置注册消息的 AUTHR 字段为鉴权算法输出的 18 比特数据；
- 通过鉴权响应消息，向基站发送鉴权数据(AUTHR)、随机查询值(RANDC：RAND 最高位的 8 个比特)以及呼叫历史参数(COUNT)。

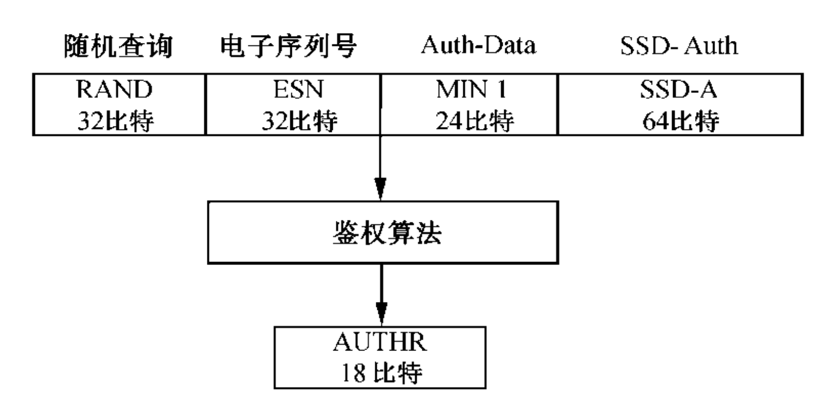


图 8.5 移动台注册鉴权的 AUTHR 计算

基站：

- 将收到的 RANDC 值与内部存储的 RAND 最高位的 8 个比特进行比较；
- 结合收到的 MSIN/ESN，将收到的 COUNT 值与它内部存储的值进行比较；
- 采取与移动台一样的方式计算出 AUTHR 值，除非使用内部存储的 SSD-A 值；
- 将计算得到的 AUTHR 值与从移动台收到的 AUTHR 值进行比较。

如果其中任何一个比较失败，基站将认为注册尝试不成功，启动惟一查询响应过程，或者开始 SSD 更新。

8.4.1 公共鉴权字段

大多数接入信道消息共享下列 4 个相同的与鉴权相关的字段：

1. 鉴权模式(AUTH-MODE)

如果基站要求进行鉴权而且鉴权信息是可用的，则基站将这个字段设置为 01，其他值保留。

2. 鉴权数据(AUTHR)

若鉴权模式字段设为 01,移动台将设置 AUTHR 这个字段,详细说明见 8.3 节。若鉴权模式字段设为其他任何值,则基站将忽略这个字段。

3. 随机查询值(RANDC)

若鉴权模式字段设为 01,移动台将设置 RANDC 这个字段,详细说明见 8.3 节。若鉴权模式字段设为其他任何值,则基站将忽略这个字段。

4. 呼叫历史参数(COUNT)

若鉴权模式字段设为 01,移动台将把这个字段设为存储在半永久存储器中的 COUNT 参数的当前值。若鉴权模式字段设为其他任何值,则基站将忽略这个字段。

8.4.2　注册消息

这一节将详细介绍可以在接入信道上发送的注册消息体的内容。当移动台发送了一个注册消息,它将会用到下列变长消息格式:

字　　段	长度(比特)	字　　段	长度(比特)
消息类型(00000001)	8	鉴权模式	2
确认序列号	3	鉴权数据	0 或 18
消息序列号	3	随机查询值	0 或 8
要求确认指示	1	呼叫历史参数	0 或 6
有效确认指示	1	注册类型	4
确认地址类型	3	时隙循环索引	3
移动台标识符字段类型	3	移动台协议修正	8
移动台标识符字段长度	4	站级标志	8
移动台标识符	8 × MSID − LEN	移动终端呼叫接受指示	1
		保留比特	6

变长字段包含在注册消息体中,进一步的说明如下:

- 移动台标识符字段类型(MSID-TYPE)

与 MSID 字段中包含的标识符类型相对应,移动台将这个字段的值设置为表 8.3 中的值。

表 8.3　地址类型

说明	MSID 类型(二进制)	MSID 长度(字节数)
MSIN 和 ESN	000	9
其他所有的 MSID-TYPE 值保留		

- 移动台标识符字段长度(MSID-LEN)
- 移动台将这个字段设为 MSID 字段中的字节数
- 移动台标识符(MSID)

利用 MSID-TYPE 字段中指出的标识符类型,移动台将这个字段设为移动台的标识符。若 MSID-TYPE 等于 000,则 MSID 字段包含下列子字段:

子字段	长度(比特)
移动台识别号 1(MSIN1)	24

移动台识别号 2(MSIN2)	10
移动台的电子序列号(ESN)	32
保留(000000)	6

8.4.3 接入信道消息体格式

在接入信道上发送的消息总结如表 8.4 所示。

表 8.4 接入信道消息

消息名	消息类型(二进制)
注册消息	00000001
指令消息	00000010
数据突发消息	00000011
始发消息	00000100
寻呼响应消息	00000101
鉴权查询响应消息	00000110

8.5 惟一查询响应过程

惟一查询响应过程是由基站启动的,并且它既可以在寻呼信道和接入信道上实现也可以在前向和反向业务信道上实现。过程如下:

基站:

- 生成 24 比特的随机查询数据(RANDU),并且在前向业务信道的寻呼信道上通过惟一(或鉴权)查询消息将它发往移动台(W-CDMA 中,前向业务信道的信令信道);
- 运行鉴权算法,如图 8.6 所示;
- 将 AUTHU(鉴权查询响应)设置成 18 比特的鉴权输出。

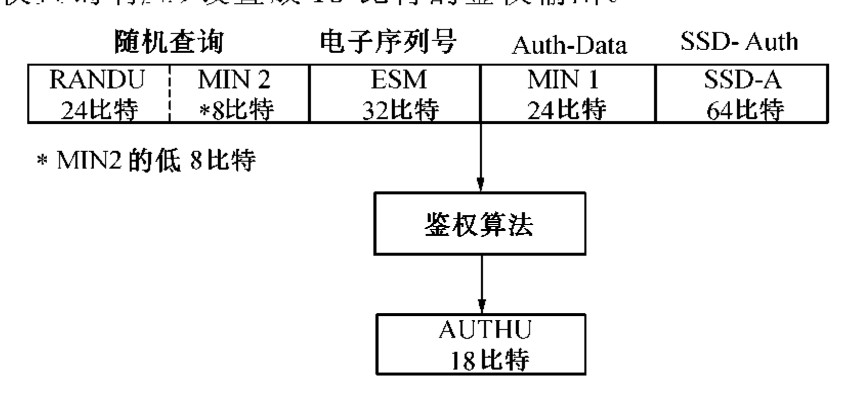


图 8.6 惟一查询响应过程的 AUTHU 计算

移动台收到惟一查询请求消息后:

- 设置鉴权算法的输入参数,如图 8.6 所示;
- 设置存储注册输入参数为 FALSE;
- 按上面的方法用接收到的 RANDU 计算出 AUTHU,其余参数用内部存储值;

- 通过惟一查询响应消息将 AUTHU 发给基站。

收到移动台的惟一查询响应后，基站将收到的 AUTHU 值与本身产生的 AUTHU 值或者内部存储值做比较，如果比较失败，基站将拒绝移动台进一步的接入尝试，终止进行中的呼叫，或者启动共享加密数据（SSD）更新过程。

8.5.1　鉴权查询消息

当基站在寻呼信道上发送鉴权查询消息时，它使用下面的定长消息格式：

字　段	长度（比特）
消息类型（MSG-TYPE）00001010	8
确认序列号（ACK-SEQ）	3
消息序列号（MSG-SEQ）	3
要求确认指示（ACK-REQ）	1
有效确认指示（VALID-ACK）	1
地址类型（ADDR-TYPE）	3
地址字段长度（ADDR-LEN）	4
移动台地址（ADDRESS）	8 × ADDR – LEN
随机查询数据（RANDU）	24
保留比特（RESERVED）000	3

8.5.2　鉴权查询响应消息

当移动台在接入信道上发送鉴权查询响应消息时，它使用下面的定长消息格式：

字　段	长度（比特）
消息类型（MSG-TYPE）00000110	8
确认序列号（ACK-SEQ）	3
消息序列号（MSG-SEQ）	3
要求确认指示（ACK-REQ）	1
有效确认指示（VALID-ACK）	1
确认地址类型（ACK-TYPE）	3
移动台标识符字段类型（MSID-TYPE）	3
移动台标识符字段长度（MSID-LEN）	4
移动台标识符（MSID）	8 × MSID-LEN
保留比特（RESERVED）	2
这些比特代替 AUTH-MODE 字段	
移动台将这个字段设为 00	
鉴权查询响应（AUTHU）	18
保留比特（RESERVED）0000	4

8.6 移动台发起呼叫鉴权

当鉴权标识信息单元上的鉴权指示设为 01，并且移动台尝试通过在接入信道上发送始发消息发起一个呼叫时，与鉴权相关的过程将会执行如下：

移动台：

- 设置鉴权算法的输入参数，如图 8.7 所示；
- 设置存储注册输入参数为 TRUE；
- 运行鉴权算法；
- 设置鉴权响应（AUTHR）的值等于鉴权算法的输出（18 比特）；
- 通过鉴权响应消息，向基站发送 AUTHR，RANDC（RAND 最高位的 8 个比特）和 COUNT。

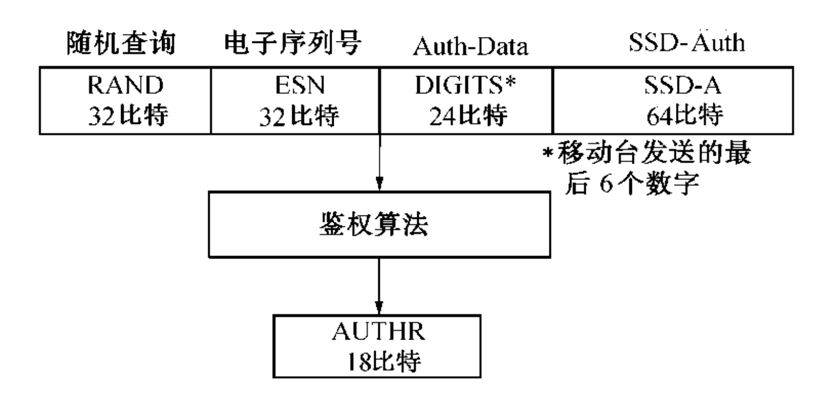


图 8.7 移动台发起呼叫鉴权的 AUTHR 计算

基站：

- 将收到的 RANDC 值与内部存储的 RAND 最高位的 8 个比特进行比较；
- 结合收到的 MSIN/ESN，将收到的 COUNT 值与它内部存储的值进行比较；
- 使用其内部的 SSD-A，采取与移动台一样的方式计算出 AUTHR 值；
- 将计算得到的 AUTHR 值与从移动台收到的 AUTHR 值进行比较。

如果比较成功，则启动相应的信道指配程序，一旦指配了一个前向业务信道，基站将会根据系统的判断发出一个参数更新指令给移动台，用来更新存储在移动台半永久存储器中的 COUNT 的值。

如果其中任何一个比较失败，基站将拒绝服务，启动惟一查询响应过程，或者开始 SSD 更新。

8.7 移动台中断鉴权

如果 AUTH（鉴权模式，2 比特）的值为 01，则在接入信道消息中适当的位置会加入鉴权数据。

当在寻呼信道上发送的接入参数消息 AUTH 字段值设为 01（标准鉴权模式），并且移动台响应寻呼时（在接入信道上发送寻呼响应消息），执行下面的鉴权过程：

移动台：

- 设置鉴权算法的输入参数，如图 8.8 所示；
- 设置存储注册输入参数为 TRUE；

- 运行鉴权算法；
- 设置 AUTHR 的值等于鉴权算法的输出（18 比特），用来填充寻呼响应消息的 AUTHR 字段；
- 通过寻呼（鉴权）响应消息，向基站发送 AUTHR，RANDC（RAND 最高位的 8 个比特）和 COUNT。

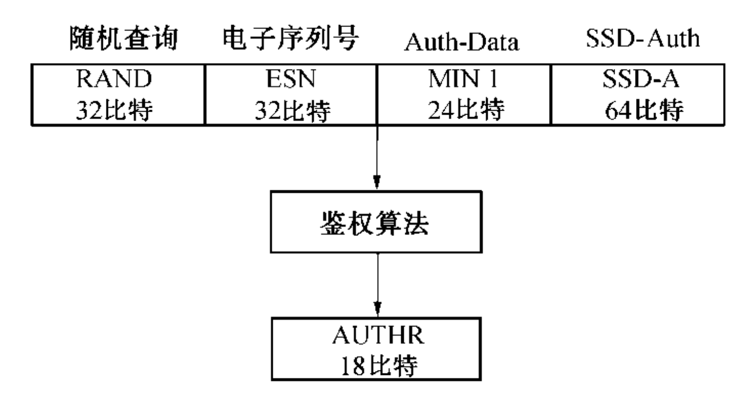


图 8.8　移动台终止呼叫鉴权的 AUTHR 计算

基站：

- 将收到的 RANDC 值与内部存储的 RAND 最高位的 8 个比特进行比较；
- 结合收到的 MSIN/ESN，将收到的 COUNT 值与它内部存储的值进行比较；
- 使用其内部的 SSD-A，采取与移动台一样的方式计算出 AUTHR 值；
- 将计算得到的 AUTHR 值与从移动台收到的 AUTHR 值进行比较。

如果比较成功，则启动相应的信道指配程序，一旦指配了一个前向业务信道，基站将发出一个参数更新指令给移动台，用来更新存储在移动台半永久存储器中的 COUNT 的值。

如果其中任何一个比较失败，基站将拒绝服务，启动惟一查询响应过程，或者开始 SSD 更新。

接入参数消息定义的参数是当移动台在接入信道上给基站发送信息时移动台用到的。读者参考时，注意：（1）如果移动台要在接入信道消息中包含标准鉴权数据，则基站必须把这个字段设为 01，如果移动台不想在接入信道消息中包含鉴权数据，则基站把这个字段设为 00；（2）如果 AUTH 字段值为 01，则基站把这个字段设为随机查询值（RAND），移动台用它来进行鉴权，如果 AUTH 字段设为其他值，基站将忽略这个字段。

8.8　SSD 更新

SSD 的更新是由 SSD 生成程序进行的，并且用移动台特殊信息、随机数及移动台的 A 钥进行初始化。

A 钥是：

- 64 比特长；
- 分配给移动台的；
- 存储在移动台永久性安全标识存储器中；
- 仅对移动台和相关的归属位置寄存器/鉴权中心（HLR/AC）是可知的。

注意：

1. 上面最后一项增强了移动台加密数据的安全性，这是因为用户漫游时不需要将 A 钥从一个基站传到另一个基站。因此，SSD 更新只在移动台和相关的 HLR/AC 中执行，而不是在正在服务的基站中执行，正在服务的基站通过与移动台的 HLR/AC 进行内部系统通信得到一份由 HLR/AC 计算出的 SSD 副本。
2. 既然 SSD 的更新过程包括多事务处理，且可以在一个信道上开始在另一个信道上完成，那么为了额外的透明度，呼叫处理和信令文本也必须包含在这里。

A 钥必须成为移动台的一部分，更进一步的移动台中 SSD 更新的图解说明见图 8.9。

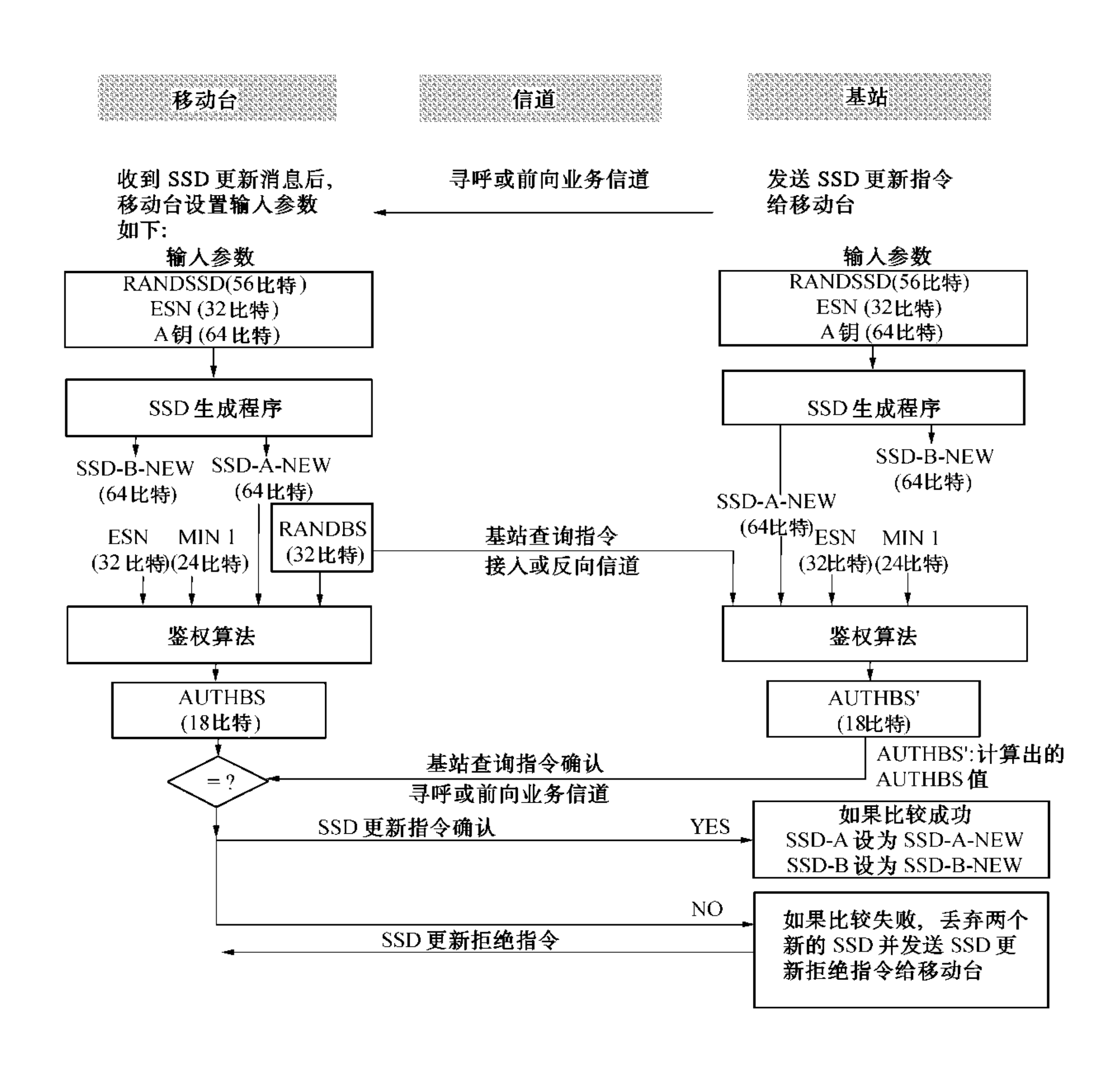


图 8.9　SSD 更新消息流

SSD 更新过程执行如下：

基站：

- 通过寻呼信道或前向业务信道的 SSD 更新消息向移动台发送 SSD 更新指令，其中 RANDSSD 字段中包含的 56 比特随机数与 HLR/AC 中计算的相同。

移动台收到 SSD 更新消息后：

- 设置 SSD 生成程序的输入参数；
- 运行 SSD 生成程序；
- SSD 发生器的输出结果的高位 64 比特为 SSD-A-NEW，低位 64 比特为 SSD-B-NEW（见图 8.10）；
- 如图 8.9 所示，设置鉴权算法的输入参数；
- 执行鉴权算法；
- 选择一个 32 比特随机数，RANDBS，通过接入信道或反向业务信道上的基站查询指令发往基站；
- 设置 AUTHBS 等于鉴权算法输出的 18 比特；
- 设置存储注册输入参数为 FALSE。

基站收到基站查询消息后：

- 如图 8.9 所示，设置鉴权算法的输入参数，其中 RANDBS 设为基站查询消息收到的值；
- 设置存储注册输入参数为 FALSE；
- 执行鉴权算法；
- 设置 AUTHBS 等于鉴权算法输出的 18 比特（见图 8.11）；
- 证实收到基站查询指令，这是通过在寻呼信道或前向业务信道的基站查询确认指令中包含 AUTHBS 实现的，也就是说，在寻呼信道或前向业务信道的基站查询确认指令中，将它计算的 AUTHBS 值发往移动台。

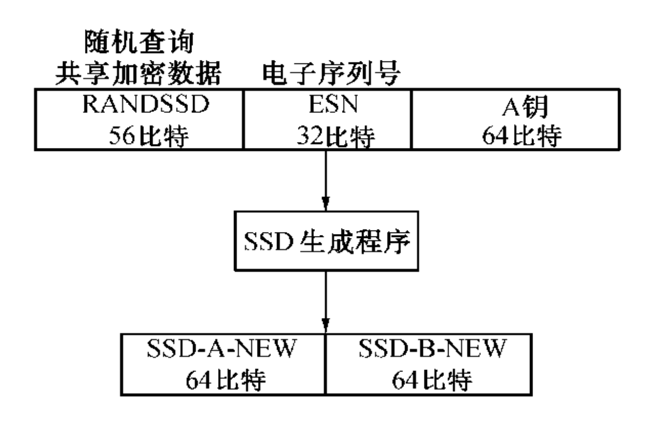


图 8.10　共享加密数据（SSD）的计算

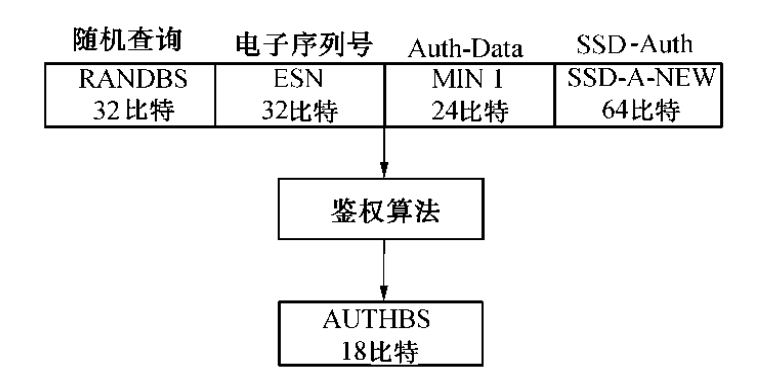


图 8.11　AUTHBS 的计算

移动台收到基站查询确认指令后：

- 比较收到的 AUTHBS 的值与移动台自己计算出的值；
- 证实收到 SSD 更新指令确认，过程如下：

1. 如果移动台比较成功，则执行 SSD 更新程序，将 SSD-A 更新为 SSD-A-NEW，SSD-B 更新为 SSD-B-NEW；
2. 发送 SSD 更新确认消息到基站，用来指示 SSD 更新的成功完成，其中 SSD 包含在鉴权参数信息单元中。
3. 如果比较失败，则丢弃 SSD-A-NEW 和 SSD-B-NEW；

4. 发送 SSD 更新确认消息给基站，用来指示 SSD 更新没有成功完成，其中 SSD 包含在鉴权参数信息单元中。

在基站中，如果从移动台收到的 SSD 更新指令确认消息说明更新成功，则将 SSD-A 和 SSD-B 的值设为从 HLR/AC 收到的值。

在确定基站肯定收到基站查询指令之后的特定时间内，如果移动台还收不到基站查询指令的确认消息，则移动台将丢弃 SSD-A-NEW 和 SSD-B-NEW，并终止 SSD 更新过程。

8.9 信令消息加密

为了加强鉴权过程和保护用户的敏感信息（比如 PINS），一种有效的方法是对所选业务信道中信令消息的某些字段进行加密，然而，TIA/EIA/IS-95 既没有讨论也没有列出要加密的消息和字段，原因是加密算法的使用完全受控于美国国际交易和武器规则（ITAR）和出口管理规则。

如果鉴权不执行（即接入参数消息的 AUTH 字段值为 00），将不对消息进行加密。

信令消息的加密是由每个呼叫单独控制的，信道指配时通过加密消息中信令加密字段的值来设定呼叫的初始加密模式。如果信令加密字段设为 00，则不进行消息加密，如果想在信道指配后打开加密模式，基站将要发送一个信令加密字段值为 01 的加密消息；如果想关闭加密模式，基站将要发送一个信令加密字段值为 00 的加密消息。

每个反向业务信道的消息中包含一个加密字段，用来识别消息创建时激活的加密模式。

8.10 话音保密

在 CDMA 系统中，话音保密是通过采用专用长码掩码进行 PN 扩频来实现的，专用长码掩码的生成和应用在 TIA/EIA/IS-95 中没有特别指定。

系统仅在业务信道上提供话音加密，所有呼叫初始化时都采用公用长码掩码进行 PN 扩频。在呼叫建立期间，移动用户通过话音保密字段设为 01 的始发消息或寻呼响应消息请求话音保密，而在通话期间，移动台使用话音保密字段设为 01 的加密消息请求话音保密。

如果鉴权过程不执行（接入参数消息中的 AUTH 字段设为 00 或者移动台不能执行鉴权），则不会执行公用到专用长码掩码的转换。

要启动公用与专用长码掩码之间的转换，基站或移动台必须在业务信道上发送一个长码转换请求指令。

移动台对长码转换请求指令响应的具体说明如下：

- 如果转换请求指令是要求转换到专用长码，并且移动台接收这个请求，那么移动台将在 $T_{56m}=0.2$ 秒内发送一个长码转换响应指令（00000011），移动台将在前向和反向业务信道上都使用专用长码。如果使用时间等于 0，在响应发送结束 $N_{4m}=20$ 帧后的第一个 80 ms 边界处（相对于系统时间的起点），移动台将开始使用专用长码，并提示用户话音保密模式被激活。如果转换请求指令要求转换到专用长码，而移动台不能生成专用长码或移动台不接收这个请求，则它将在 T_{56m} 秒内发送一个长码转换响应指令（00000010）。
- 如果转换请求指令是要求转换到公用长码，并且移动台接收这个请求，那么移动台将在 T_{56m}

秒内发送一个长码转换响应指令（00000010），并提示用户话音保密模式没有被激活。如果转换请求指令要求转换到公用长码，而移动台不接收这个请求，则它将在 T_{56m} 秒内发送一个长码转换响应指令（00000011）。

基站对长码转换请求指令响应的具体说明在下面介绍，如果要求话音保密的请求在始发消息或寻呼响应消息中指定，基站可以发送一个长码转换请求指令（00000001）要求转换到专用长码。

基站处理长码转换请求指令的过程如下：

- 如果转换请求指令是要求转换到专用长码，并且基站接收这个请求，那么基站将发送一个长码转换请求指令（00000001），如果基站不接收这个请求，那么基站将发送一个长码转换请求指令（00000000）。
- 如果转换请求指令是要求转换到公用长码，并且基站接收这个请求，那么基站将发送一个长码转换请求指令（00000000），如果基站不接收这个请求，那么基站将发送一个长码转换请求指令（00000001）。

基站处理长码转换响应指令的过程如下：[1]

- 如果长码转换响应指令表示移动台接收基站发送的长码转换请求指令中要求的长码转换，则基站将在前向和反向的业务信道上都采用要求的长码掩码，如果基站没有在转换请求中规定明确的作用时间，在长码转换响应指令的最后一帧被完全接收后，再过 N_{4m} 帧的第一个 80 ms 边界处（相对于系统时间的起点），基站将开始使用要求的长码掩码。

8.11　鉴权算法

在计算机通信网络中，通常必须对通信各部分的身份进行验证，实用的一种方法是采用加密鉴权协议，这种协议使用单向 Hash 函数实现。为了验证移动台的身份，移动台必须与基站协同操作，CDMA 系统中，通过在移动台与基站之间交换信息来确认移动台的身份的过程就是鉴权过程。

考虑将一种可能的鉴权方案用于任何分组密码的情形中，采用一种对称分组密码算法（比如 DES）来计算 18 比特 Hash 值。如果分组算法是安全的，那么可以假定单向 Hash 函数也是安全的。

152 比特的消息块 M 输入到鉴权算法设备中，利用 DES，M 首先被分成 64 比特长的块，这样 $M = M_1M_2M_3\cdots$。第一个信息块 M_1 成为 DES 密钥。通过对 152 比特消息填充 40 比特 0 后将它映射成 192 比特，这样便完成了 64 比特块的划分。适当的填充是为了方便将消息分成某些固定长度的块。这里的鉴权方案就是从 192 比特长的消息中生成 18 比特的鉴权数据。

对于移动台的注册、惟一查询响应过程、发起呼叫、中断和 SSD 更新这些过程中得到的一些可信的计算结果，我们将在下面依次讨论。

这节中还将提出一些关于密钥生成方面的方法，很遗憾，密钥生成方法的设计并不像人们从易实现且安全的 Hash 函数观点来看时所期望的那么容易。接下来，做了一些尝试来设计密钥生成结构，见图 8.12 的灰色块。

---

[1] 此处原文为请求指令，可能有误——译者注。

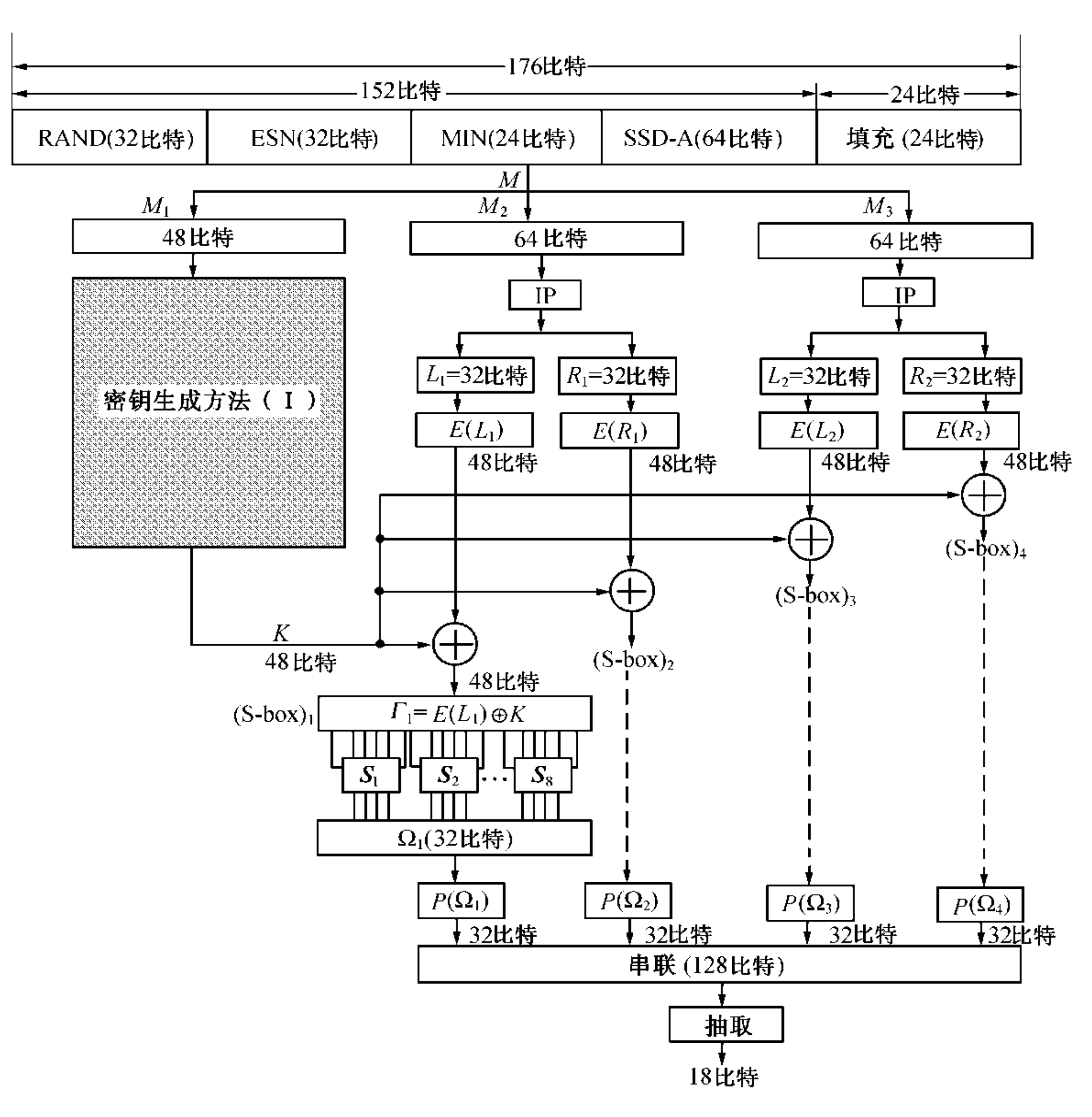


图 8.12　移动台注册或终止呼叫的 AUTHR(18 比特)计算

8.11.1　密钥生成技术(Ⅰ)和 AUTHR 的计算

假定 M_1, M_2, M_3 分别是 176 比特分解后得到的 48 比特、64 比特和 64 比特，176 比特值由 152 比特长的消息和 24 比特的填充比特组成。$M_1 = 48$ 比特将用做密钥生成方案的输入(即图 8.12 中的灰色块)。

将 48 比特的输入排列成 6×8 的阵列，如下所示：

表 8.5　用于密钥计算的6×8阵列

输入(按列)

1	7	13	19	25	31	37	43
2	8	14	20	26	32	38	44
3	9	15	21	27	33	39	45
4	10	16	22	28	34	40	46
5	11	17	23	29	35	41	47
6	12	18	24	30	36	42	48

列向置换

7	31	1	43	13	37	25	19
8	32	2	44	14	38	26	20
9	33	3	45	15	39	27	21
10	34	4	46	16	40	28	22
11	35	5	47	17	41	29	23
12	36	6	48	18	42	30	24

行向置换

11	35	5	47	17	41	29	23	
7	31	1	43	13	37	25	19	
9	33	3	45	15	39	27	21	⇨ 输出(按行)
12	36	6	48	18	42	30	24	
8	32	2	44	14	38	26	20	
10	34	4	46	16	40	28	22	

从 M_1 生成 48 比特密钥的计算如下所示:

11	35	5	47	17	41	29	23	7	31	1	43	13	37	25	19
9	33	3	45	15	39	27	21	12	36	6	48	18	42	30	24
8	32	2	44	14	38	26	20	10	34	4	46	16	40	28	22

例 8.2　首先,考虑用上表中的列/行置换来生成密钥。假定密钥数据是16c27a415f39(十六进制) = 0001 0110 1100 0010 0111 1010 0100 0001 0101 1111 0011 1001(二进制)。

参照表 8.5,密钥数据的运行情况如下:

1. 按列输入密钥数据

⇩

0101 0011
0001 1111
0111 0011
1100 0110
0001 0000
1010 0101

2. 执行列置换

1001 0101

0101 0111

1001 1101

1110 0100

0000 0001

0111 1000

3. 执行行置换

0000 0001

1001 0101

1001 1101 ⇩ 01959d7857e4

0111 1000

0101 0111

1110 0100

这样，得到的 48 比特密钥按行书写如下：

0000 0001 1001 0101 1001 1101 0111 1000 0101 0111 1110 0100（二进制） = 01959d7857e4（十六进制）

通过执行密钥生成技术（I），计算出 48 比特密钥 K，它可以应用于图 8.12 中的灰色块。

例 8.3 现在，使用图8.12，从 Hash 变换后的 176 比特已填充消息中得到 18 比特的 AUTHR 消息摘要的过程如下：

假定 M_2 数据块（64 比特）是 17b439a12f51c5a8，对 M_2 数据首先进行一个初始置换使它分成两个块 L_1（左）和 R_1（右），其中每个块都是 32 比特，如表 8.6 所示。

先把 M_2 十六进制的数转换成二进制值，然后按表 8.6 排列，初始置换后的序列可以通过下面的过程得到：给定 M_2 如下

17b439a1→0001 0111 1011 0100 0011 1001 1010 0001

2f51c5a8→0010 1111 0101 0001 1100 0101 1010 1000

根据表 8.6 执行初始置换：

表 8.6 初始置换（**IP**）

	58	50	42	34	26	18	10	2
L_1	60	52	44	36	28	20	12	4
	62	54	46	38	30	22	14	6
	64	56	48	40	32	24	16	8
	57	49	41	33	25	17	9	1
R_1	59	51	43	35	27	19	11	3
	61	53	45	37	29	21	13	5
	63	55	47	39	31	23	15	7

L_1(32 比特)	0110	0000	0010	0111	0101	0011	0111	1101
	6	0	2	7	5	3	7	d
R_1(32 比特)	1100	1010	1001	1110	1001	0100	0001	0001
	c	a	9	e	9	4	1	1

这样，初始置换后的表示如下：

6027537d(L_1)；ca9e9411(R_1)

如上所示 64 比特置换后的序列包含两个 32 比特的块 L_1 和 R_1。

根据表 8.7 分别将 L_1 和 R_1 从 32 比特扩展到 48 比特。

表 8.7　比特扩展表

32	1	2	3	4	5
4	5	6	7	8	9
5	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

$E(L_1)$表示这样一个函数，即输入 32 比特的数据块，输出 48 比特的数据块，如下所示：

$$E(L_1) = \mathrm{b\ 0\ 0\ 1\ 0\ e\ a\ a\ 6\ b\ f\ a}$$

$E(L_1)$算出后，将它与密钥 K 逐位相加

$$\begin{aligned}\Gamma_1 &= E(L_1) \oplus K \\ &= (\mathrm{b0010eaa6bfa}) \oplus (\mathrm{01959d7857e4}) \\ &= (\mathrm{b19493d23c1e})\end{aligned}$$

将 48 比特的 Γ_1 输入到(S-box)$_1$ 是通过一个非线性的置换得到 32 比特的输出来实现的。

48 比特的矢量 Γ_1 用做(S-box)$_1$ 从 S_1 到 S_8 置换操作的一个变量，每个 S_i，$1 \leqslant i \leqslant 8$，是个 4 行 16 列的矩阵，如表 8.8 所示，每个 S_i 的输入是 6 比特，它的第一个和最后一个比特代表 S_i 的行号，中间的 4 个比特产生一个列号。比如，输入(010011)到 S_i，表示为 S_i^{01} 1001，行号是 01(即行 1)，列号由(1001)决定(即列 9)。

表 8.8　基本 S-box 函数

S_1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

（续表）

							S_2								
15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
							S_3								
10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
							S_4								
7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
							S_5								
2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
							S_6								
12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
							S_7								
4	11	2	14	15	0	8	13	2	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
							S_8								
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

若 S-box 的输入为 Γ_1 = b 1 9 4 9 3 d 2 3 c 1 e,则它的二进制表示是

101100011001010010010011110100100011110000011110

将这个 48 比特分成 6 比特一组，那么 S_1 到 S_8 置换操作的计算将会变得容易些，如下所示：

$$S_1^{10}(0110) = S_1^2(6) = 2 = 0010$$
$$S_2^{01}(1100) = S_2^1(12) = 6 = 0110$$
$$S_3^{00}(1001) = S_3^0(9) = d = 1101$$
$$S_4^{01}(1001) = S_4^1(9) = 7 = 0111$$
$$S_5^{10}(1010) = S_5^2(10) = c = 1100$$
$$S_6^{11}(0001) = S_6^3(1) = 3 = 0011$$
$$S_7^{10}(1000) = S_7^2(8) = a = 1010$$
$$S_8^{00}(1111) = S_8^0(15) = 7 = 0111$$

将所有这些 4 比特的 S_i，$1 \leqslant i \leqslant 8$，串联起来，就得到 32 比特的输出 Ω_1：

$$\Omega_1 = 0010\ 0110\ 1101\ 0111\ 1100\ 0011\ 1010\ 0111 = 26d7c3a7$$

这就是 $(S\text{-box})_1$ 的输出。

使用表 8.9，32 比特置换后的输出 $P(\Omega_1)$ 是通过将 Ω_1 的比特按下面的方法置换后产生的：Ω_1 的第 16 个比特作为 $P(\Omega_1)$ 的第 1 个比特，Ω_1 的第 7 个比特作为 $P(\Omega_1)$ 的第 2 个比特，就这样直到 Ω_1 的第 25 个比特作为 $P(\Omega_1)$ 的第 32 个比特。

表 8.9　置换函数 P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

这 $P(\Omega_1)$ 的结果表示为

$$P(\Omega_1) = 1100\ 0101\ 0110\ 0111\ 0011\ 1111\ 0011\ 0001 = c5673f31$$

接下来，考虑 M_2 的右 32 比特 R_1。若 R_1 使用表 8.7 扩展到 48 比特，可以得出

$$E(R_1) = e554fd4a80a3$$

$E(R_1)$ 和 K 进行异或操作得到 48 比特 Γ_2，结果输入到 $(\text{S-box})_2$ 中进行计算，如下所示：

$$\begin{aligned}\Gamma_2 &= E(R_1) \oplus K \\ &= (e554fd4a80a3) \oplus (01959d7857e4) \\ &= e4c16032d747 \\ &= 1110\ 0100\ 1100\ 0001\ 0110\ 0000\ 0011\ 0010\ 1101\ 0111\ 0100\ 0111\end{aligned}$$

Γ_2 分成 8 组，每组 6 比特，$(\text{S-box})_2$ 的输出如下所示：

i	输入到 S_i	S_i，$1 \leqslant i \leqslant 8$		S_i 的输出
	Γ_2	行	列	Ω_2

1	111001	3	12	a
2	001100	0	6	3
3	000101	1	2	0
4	100000	2	0	a
5	001100	0	6	b
6	101101	3	6	f
7	011101	1	14	8
8	000111	1	3	8

因而，(S-box)$_2$ 的输出是

$$\Omega_2 = a30abf88$$

使用表 8.9 置换后的序列为

$$P(\Omega_2) = 79c062c9$$

因此现在已经完全计算出了对应于 64 比特 M_2 数据块的置换数据 $P(\Omega_1)$和 $P(\Omega_2)$。

假定第三个数据块 M_3(64 比特)是 51cb36af43000000，使用表 8.6，将 M_3 进行初始置换后的数据序列为 13050c1b0a0c0a1e，这里 L_2 = 13050c1b，R_2 = 0a0c0a1e，根据表 8.7，将 M_3 的 L_2(左半)和 R_2(右半)从 32 比特扩展到 48 比特，如下所示：

$$E(L_2) = 8a680a8580f6$$

可以看到，32 比特的 L_2 或 R_2 可以用比特扩展表扩展并扰码成 48 比特。

扩展后的数据 $E(L_2)$或 $E(R_2)$与密钥 K 进行异或运算：

$$\begin{aligned}\Gamma_3 &= E(L_2)\oplus K\\ &= (8a680a8580f6)\oplus(01959d7857e4)\\ &= (8bfd97fdd712)\\ &= (1000\ 1011\ 1111\ 1101\ 1001\ 0111\ 1111\ 1101\ 1101\ 0111\ 0001\ 0010)\end{aligned}$$

将 48 比特 Γ_3 输入到(S-box)$_3$ 中。

$\boldsymbol{S}_1$ 到 $\boldsymbol{S}_8$ 的(S-box)$_3$ 的运算如下：

输入到 S_i	$S_i, 1 \leqslant i \leqslant 8$		S_i 的输出
Γ_3	行	列	Ω_3
S_1^{10}	2	1	1
S_2^{11}	3	15	9
S_3^{10}	2	11	12
S_4^{01}	1	11	12
S_5^{11}	3	15	3
S_6^{01}	1	14	3
S_7^{00}	0	14	6
S_8^{00}	0	9	9

(S-box)$_3$ 的输出数据可以写成

$$\Omega_3 = 19cc3369$$

置换函数 $P(\Omega_3)$依据表 8.9 将输入的 32 比特 Ω_3 置换后输出 32 比特的数据。

得到置换后 $P(\Omega_3)$的值

$$P(\Omega_3) = 28397dc2$$

最后，考虑 64 比特 M_3 初始置换(表 8.6)后的右边的一半 R_2 = 0a0c0a1e，将它从 32 比特扩展

到 48 比特，得到 $E(R_2)$ 为

$$E(R_2) = 0540580540fc$$

$(\text{S-box})_4$ 的输入数据 Γ_4 是通过由 $E(R_2)$ 与 K 进行异或运算得到的

$$\Gamma_4 = E(R_2) \oplus K$$
$$= 04d5c57d1718$$

其中 K 为密钥数据序列。

$\boldsymbol{S}_1$ 到 $\boldsymbol{S}_8$ 的 $(\text{S-box})_4$ 的运算如下：

输入到 S_i	$S_i, 1 \leqslant i \leqslant 8$		S_i 的输出
Γ_4	行	列	Ω_4
S_1^{10}	1	0	0
S_2^{01}	1	6	8
S_3^{01}	1	11	14
S_4^{01}	1	2	11
S_5^{01}	1	15	6
S_6^{01}	1	8	6
S_7^{00}	0	14	6
S_8^{00}	0	12	5

计算出的 $(\text{S-box})_4$ 的输出 Ω_4 为

$$\Omega_4 = 08eb6665$$

使用表 8.9，Ω_4 置换后得到

$$P(\Omega_4) = 807d0dec$$

这样，我们已经完全计算出所有 4 个置换值，即 $P(\Omega_1)$，$P(\Omega_2)$，$P(\Omega_3)$ 和 $P(\Omega_4)$，最后输出的 128 比特 $P(\Omega)$ 可以通过下面的串联过程得到：

$$P(\Omega) = P(\Omega_1) \parallel P(\Omega_2) \parallel P(\Omega_2) \parallel P(\Omega_4)$$

$$P(\Omega) = (c5673f31, 79e062c9, 28397dc2, 807d0dec)$$
$$= (1100\ 0101\ 0110\ 0111\ 0011\ 1111\ 0011\ 0001$$
$$0111\ 1001\ 1110\ 0000\ 0111\ 0010\ 1100\ 1001$$
$$0010\ 1000\ 0011\ 1001\ 0111\ 1101\ 1100\ 0010$$
$$1000\ 0000\ 0111\ 1101\ 0000\ 1101\ 1110\ 1100)$$

最后，采用抽取的方法从 128 比特的 $P(\Omega)$ 中每 7 个比特取 1 个比特得到 18 比特的鉴权数据（AUTHR）：

AUTHR：011111000011000101

8.11.2　密钥生成技术（II）和 AUTHR 的计算

为了把消息分成 3 个 64 比特长度的数据块，将 152 比特长度的消息填充 40 比特后扩展成 192 比特的消息，M_i = 64 比特，$1 \leqslant i \leqslant 3$。

消息 M 平均地分成 64 比特块：

$$M_1, M_2, M_3$$

这里，为了生成 48 比特的加密密钥，必须对 $M_1 = 64$ 比特的数据进行 Hash 变换。这样，第一个消息块 M_1 变成 DES 密钥，可以应用于图 8.14 中的灰色块，图 8.13 说明了用这个方法生成密钥的

过程。

用图 8.13 计算 DES 密钥,同时还有表 8.10,8.11 和 8.12,它们分别用于置换选择 1 和选择 2 以及左移位的位数。

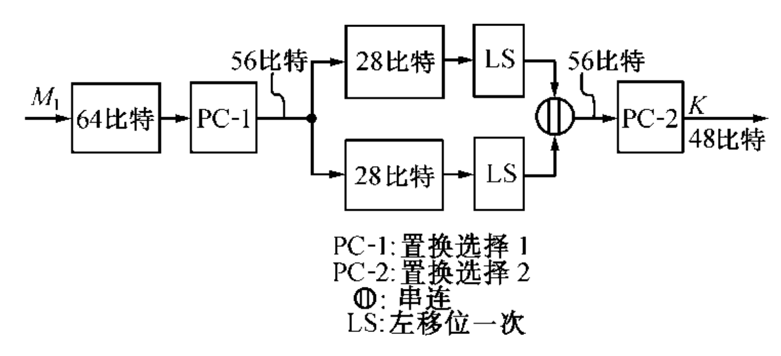


图 8.13 基于 DES 密钥方案的密钥生成图

表 8.10 置换选择1(PC-1)

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

表 8.11 置换选择 2(PC-2)

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	51	51	45	33	48
44	49	39	56	34	53	46	50	50	36	29	32

表 8.12 左移位的位数

轮号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
左移位的位数	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

例 8.4 参考图8.14,通过 Hash 变换将 64 比特的 DES 密钥 M_1 变成 48 比特密钥,过程如下:假定 DES 密钥值是 7a138b2524af17c3,依据表 8.10,PC-1 的输出为

$$L_0 = \text{a481394(左半)};R_0 = \text{e778253(右半)}$$

根据表 8.12 左移位一次(LS),输出的移位后(LS)数据为

4902729 cef04a7

串联这两个移位后的数据,再根据表 8.11 把它变换成 48 比特的密钥,这样得到的密钥数据为

$$K = \text{058c4517a7a2}$$

例 8.5 给定 M_2 = 17b439a12f51c5a8

先根据表 8.6 初始置换 M_2,然后将它分成两个 32 比特的数据块 L_1(左)和 R_1(右),得到初始置换后的数据为

$$\text{IP}(L_1) = \text{6027537d} \quad \text{IP}(R_1) = \text{ca9e9411}$$

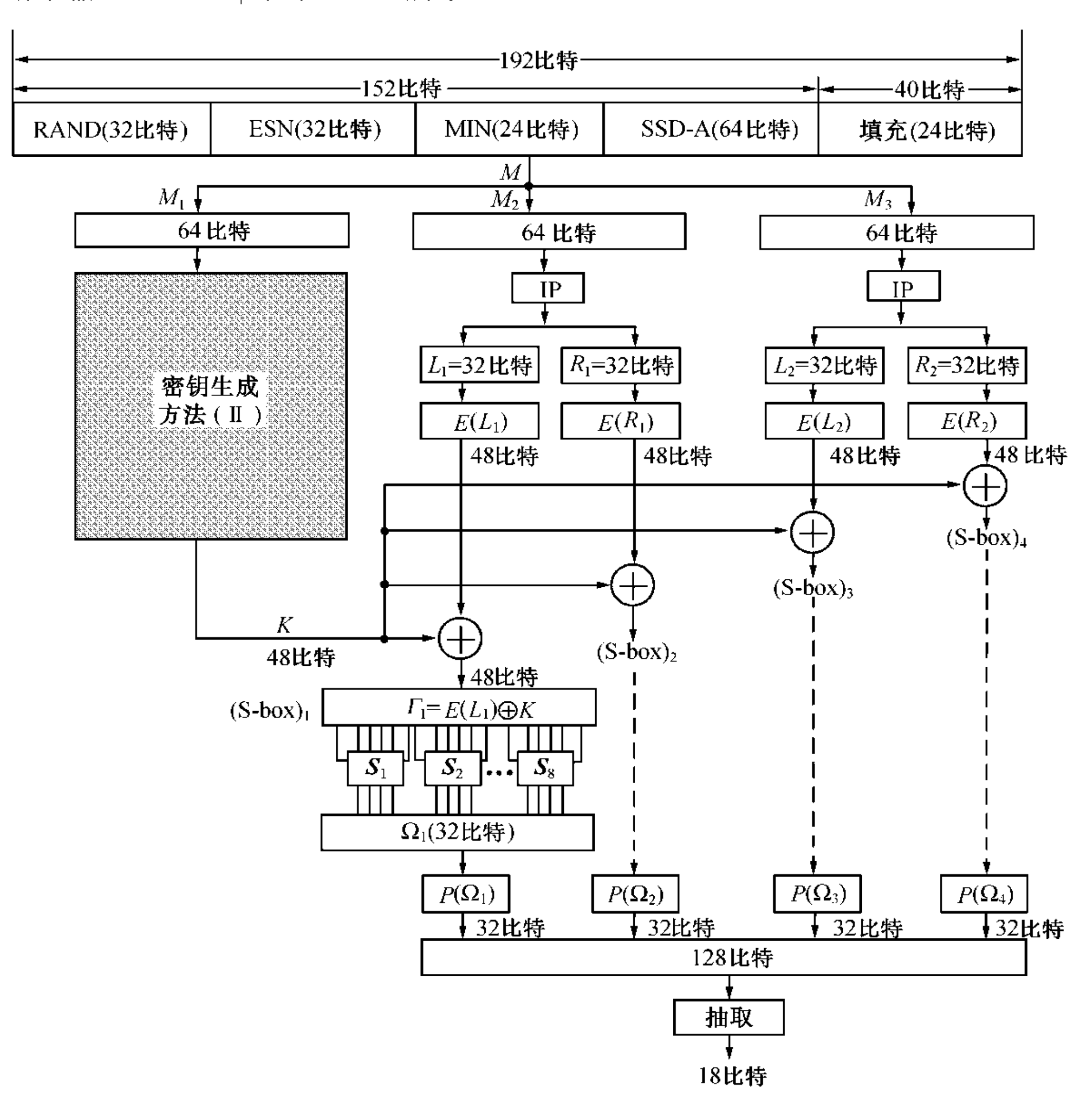
1. 使用表 8.7 进行数据扩展得到 $E(L_1)$

$E(L_1)$ = b0010eaa6bfa

$E(L_1)$和 K 进行异或运算,得到

$$\begin{aligned}\Gamma_1 &= E(L_1)\oplus K\\ &= (\mathrm{b0010eaa6bfa})\oplus(\mathrm{058c4517a7a2})\\ &= \mathrm{b58a4bbacc58}\end{aligned}$$

将结果输入$(\text{S-box})_1$，如图 8.14 所示。



鉴权输入参数：

- 注册和终止呼叫相同
- 惟一查询：RAND(32 比特) = RANDU(24 比特) + MIN2(8LSBS)
- 发起呼叫：AUTH-Data(24 比特) = 数字(移动台发送的最后 6 个数字)

图 8.14　移动台在注册、惟一查询响应过程、发起呼叫以及终止呼叫时的鉴权计算

$(\text{S-box})_1$ 的操作：

输入	S_1^{11}	S_2^{00}	S_3^{11}	S_4^{01}	S_5^{11}	S_6^{00}	S_7^{11}	S_8^{00}
行	3	0	3	1	3	0	3	0
列	6	12	10	5	7	14	8	12
输出	1	12	14	15	13	5	9	5

这样，$(\text{S-box})_1$ 的输出为 $\Omega_1 = 1\text{cefd}595$。

使用表 8.9 的置换函数，得到 $P(\Omega_1)$为

$$P(\Omega_1) = \text{a}34\text{d}397\text{f}$$

2. 使用表 8.7 数据扩展 R_1 得到 $E(R_1)$

$$E(R_1) = \text{e}554\text{fd}4\text{a}80\text{a}3$$

$E(R_1)$和 K 进行异或运算得到 Γ_2，并将它输入$(\text{S-box})_2$：

$$\begin{aligned}\Gamma_2 &= E(R_1) \oplus K \\ &= (\text{e}554\text{fd}4\text{a}80\text{a}3) \oplus (058\text{c}4517\text{a}7\text{a}2) \\ &= \text{e}0\text{d}8\text{b}85\text{d}2701\end{aligned}$$

$(\text{S-box})_2$ 的操作：

输入	S_1^{10}	S_2^{01}	S_3^{10}	S_4^{10}	S_5^{01}	S_6^{00}	S_7^{00}	S_8^{01}
行	2	1	2	2	1	0	0	1
列	12	6	1	12	11	9	14	0
输出	3	8	6	5	10	13	6	1

$(\text{S-box})_2$ 的输出为 $\Omega_2 = 3865\text{ad}61$。

使用表 8.9 的置换函数，得 $P(\Omega_2)$为

$$P(\Omega_2) = 91193\text{e}8\text{e}$$

3. 给定 $M_3 = 51\text{cb}360000000000$

使用表 8.6 对 M_3 进行初始置换可以得到

$$\text{左块 } L_2 = 03050403\text{；右块 } R_2 = 02040206$$

根据表 8.7 将这两个数据块分别扩展到 48 比特。

$$E(L_2) = 80680\text{a}808006$$

$E(L_2)$和 K 进行异或运算，并将运算结果输入到$(\text{S-box})_3$：

$$\begin{aligned}\Gamma_2 &= E(L_2) \oplus K \\ &= (80680\text{a}808006) \oplus (058\text{c}4517\text{a}7\text{a}2) \\ &= 85\text{c}44\text{f}9727\text{a}4\end{aligned}$$

$(\text{S-box})_3$ 的操作：

输入	S_1^{11}	S_2^{00}	S_3^{01}	S_4^{01}	S_5^{11}	S_6^{10}	S_7^{00}	S_8^{10}
行	3	0	1	1	3	2	0	2
列	0	15	8	7	2	9	15	2
输出	15	10	2	3	12	0	1	4

因此，$(\text{S-box})_3$ 的输出为

$$\Omega_3 = \text{fa23c914}$$

根据表 8.9 计算出置换后的 $P(\Omega_3)$

$$P(\Omega_3) = \text{c3cc8226}$$

4. 接下来，考虑 $R_2 = 02040206$ 的情况

根据比特扩展表（表 8.7）扩展 R_2，得到

$$E(R_2) = \text{00400800400c}$$

$(\text{S-box})_4$ 的输入 Γ_4 计算如下：

$$\begin{aligned}\Gamma_4 &= E(R_2) \oplus K \\ &= (\text{00400800400c}) \oplus (\text{058c4517a7a2}) \\ &= \text{05cc4d17e7ae}\end{aligned}$$

$(\text{S-box})_4$ 的操作：

输入	S_1^{01}	S_2^{00}	S_3^{11}	S_4^{01}	S_5^{01}	S_6^{10}	S_7^{00}	S_8^{10}
行	1	0	3	1	1	2	0	2
列	0	14	8	6	2	15	15	7
输出	0	5	4	0	2	6	1	2

因此，$(\text{S-box})_4$ 的输出为

$$\Omega_4 = 05402612$$

使用表 8.9 对 Ω_4 进行置换，得到

$$P(\Omega_4) = 02234098$$

这样，S-box 的输出是这四个数据块串联成的单一的 18 比特 Hash，S-box 最后的输出数据是

$$P(\Omega) = P(\Omega_1) \parallel P(\Omega_2) \parallel P(\Omega_3) \parallel P(\Omega_4) = \text{a34d397f91193e8ec3cc822602234098}$$

最后，采用抽取的方法从 128 比特的 $P(\Omega)$ 中每 7 个比特取 1 个比特得到 18 比特的鉴权数据（AUTHR）：

$$\text{AUTHR} = 111100001010100100$$

8.11.3　使用连接、排列和替换（S-box）计算 AUTHR

图 8.15 说明了另一种鉴权数据的算法，计算过程描述如下：

1. 通过添加 24 个填充比特将 152 比特的消息扩大到 176 比特；
2. 将扩大后的 176 比特消息分成三个数据块：$M_1 = 48$ 比特，$M_2 = 64$ 比特，$M_3 = 64$ 比特，其中 M_1 用做 48 比特密钥；
3. 将 48 比特的密钥分成两半：$K_l = 24$ 比特，$K_r = 24$ 比特；
4. 分别置换 M_2 和 M_3 得到 L_0 和 R_0：$L_0 = 64$ 比特，$R_0 = 64$ 比特；
5. 串联 K_l 和 L_0 得到 88 比特的数据：$L_0 \parallel K_l \rightarrow 88$ 比特；
6. 串联 $L_0 \parallel K_l$ 的结果与 R_0 得到 $L_0 \parallel K_l \parallel R_0 \rightarrow 152$ 比特；
7. 将第六步的结果 152 比特与 K_r 串联
   $L_0 \parallel K_l \parallel R_0 \parallel K_r \rightarrow 176$ 比特；
8. 将 176 比特消息分成四个部分：
   $N_1 = 32$ 比特，$N_2 = 48$ 比特，$N_3 = 48$ 比特，$N_4 = 48$ 比特；

9. 使用表 8.7 将 N_1 = 32 比特扩展成 $E(N_1)$ = 48 比特；
10. 这样得到四个等分的 48 比特数据块；
11. 将这四个数据块用做四个 S-box 的输入；
12. 将四个 S-box 的输出进行模 2 加，得到 32 比特数据；
13. 最后，用抽取比特的办法从 32 比特得到 18 比特的 AUTHR。

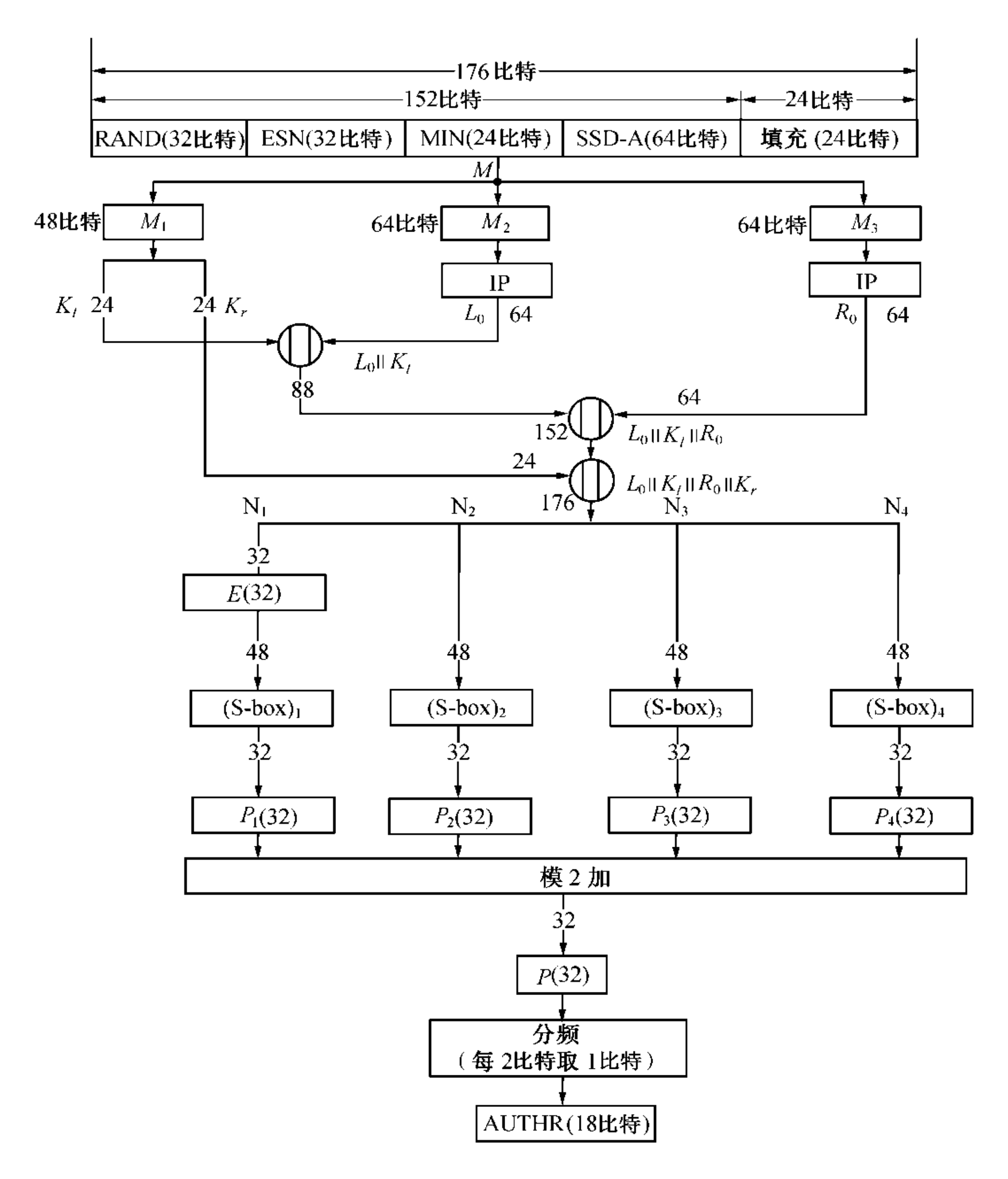


图 8.15 移动台注册或终止呼叫鉴权时 AUTHR 的计算

例 8.6 参考图8.15，将已填充的 176 比特消息分成三块：M_1 = 48 比特，M_2 = 64 比特，M_3 = 64 比特，其中，M_1 将用做密钥，K_l = 16c27a(24 比特)，K_r = 415f39(24 比特)。

假定

$$M_2 = 17b439a12f51c5a8 (64 \text{ 比特})$$

$$M_3 = 51cb36af43000000 (64 \text{ 比特})$$

使用表 8.6 分别对 M_2 和 M_3 进行初始置换，得到

$$IP(M_2) = 6027537dca9e9411 = L_0$$

$$IP(M_3) = 13050c1b0a0c0a1e = R_0$$

数据串联：$L_0 \parallel K_l \parallel R_0 \parallel K_r$（176 比特）

如下所示将串联后的数据分成四个块，即 1 个 32 比特的数据块和 3 个 48 比特的数据块：

6027537d　ca9e941116c2■7a13050c1b0a■0c0a1e415f39

|←32 比特→|←48 比特→|←48 比特→|←48 比特→|

1. 使用表 8.7 将 32 比特的数据块扩展成 48 比特

$$E(6027537d) = b0010eaa6bfa（48 比特）$$

这就是$(S\text{-box})_1$ 从 $\boldsymbol{S}_1$ 到 $\boldsymbol{S}_8$ 的输入数据，如下所示：

输入到 S_i	S_i		S_i 的输出
Γ_1	列	列	Ω_1
S_1^{10}	2	6	2
S_2^{00}	0	0	15
S_3^{00}	0	2	9
S_4^{00}	0	7	10
S_5^{10}	2	5	13
S_6^{10}	2	3	5
S_7^{11}	3	7	7
S_8^{10}	2	13	3

这样，$(S\text{-box})_1$ 的输出 $\Omega_1 = 2f9ad573$。

使用表 8.9 的置换函数，得到 $P(\Omega_1)$为

$$P(\Omega_1) = 675e6f58$$

2. $(S\text{-box})_2$ 的输入数据 $\Gamma_2 = ca9e941116c2$，写成二进制

$\Gamma_2 = 110010101001111010010100000100010001011011000010$

将 Γ_2 划分成每 6 比特一组输入到 $\boldsymbol{S}_1$ 到 $\boldsymbol{S}_8$ 中，$(S\text{-box})_2$ 输出数据的计算如下：

Γ_2	S'_i 的行和列		Ω_2
S_1^{10}	2	9	12
S_2^{11}	3	4	3
S_3^{10}	2	13	10
S_4^{00}	0	10	8
S_5^{00}	0	2	4
S_6^{01}	1	8	6
S_7^{01}	1	13	15
S_8^{00}	0	1	2

这样，$(S\text{-box})_2$ 的输出 Ω_2 变为

$$\Omega_2 = c3a846f2$$

使用表 8.9 对 Ω_2 进行置换后输出

$$P(\Omega_2)=42b6c54d$$

3.$(\text{S-box})_3$ 的输入数据 $\Gamma_3=7a13050c1b0a$。

将 Γ_3 划分成每 6 比特一组输入到$(\text{S-box})_3$ 中,得到$(\text{S-box})_3$ 输出为

$$\Omega_3=7dfbba7f$$

Ω_3 置换后输出

$$P(\Omega_3)=bf7beff6$$

4.$(\text{S-box})_4$ 的输入数据 $\Gamma_4=0c0a1e415ff39$。

用二进制表示 Γ_4,再将它分成 8 组,每组 6 比特,作为 S_i,$1\leqslant i\leqslant 8$ 的输入,将每 6 个比特输入$(\text{S-box})_4$ 中,得到$(\text{S-box})_4$ 输出如下:

$$\Omega_4=ff858d93$$

及

$$P(\Omega_4)=d38afb1b$$

因此,$(\text{S-box})_i$, $1\leqslant i\leqslant 4$ 的全部输出数据为

675e6f58 42b6c54d bf7beff6 d38afb1b

模 2 相加这四组数据,得到

49199ef8

且

$$P(49199ef8)=bf38c449$$

使用表 8.7 将置换后的数据扩展至 48 比特,如下所示:

dfe9f1608253

先将十六进制的值转换成二进制,然后分别去掉最左边和最右边的 6 比特,剩下的 36 比特序列为:

111110100111110001011000001000001001

将抽取规则应用于这个序列,每 2 个比特取 1 个比特,得到 18 比特的消息摘要(AUTHR)为:

AUTHR =(110011101100000001)

8.11.4 使用 DM 方案计算 AUTHR

DM 方案是由 Davis 和 Meyer 于 1985 年独立地提出的,它在概念上与密码分组链接(CBC)模式有某些相似之处。这一节中,我们将描述基于 DM 方案的 AUTHR 计算,采用的方法是用 DES 分组密码来构造一个 Hash 函数。首先将要进行 Hash 变换的消息分成固定长度的数据块:M_1,M_2,…,M_t,消息块 M_i,$1\leqslant i\leqslant t$ 用做加密密钥,如图 8.16 所示,前一个消息块用密钥加密后与本身异或,所得结果作为下一轮的输入,因此,到最后消息块的长度变成最后一个 M_t。$H_i(1\leqslant i\leqslant t)$计算方法为 $H_i=E_{M_i}(H_{i-1})\oplus H_{i-1}$,$1\leqslant i\leqslant t$。

采用 DES 算法，消息 M 首先被分成 56 比特的 DES 密钥块，第一个块 M_1 成为 56 比特的 DES 密钥。DES 算法的输入是一个初始向量 H_0，输出密文是 H_1，它由 $E_{M_1}(H_0)$ 与 H_0 进行异或得到，并且，H_1 成为第二个块进行 Hash 运算的输入。接下来的 56 比特消息块成为新的 DES 密钥，H_1 成为输入到算法中的明文，H_2 成为输出密文，$H_2 = E_{M_2}(H_1) \oplus H_1$，$H_2$ 为第三个块进行 Hash 运算的输入，就这样计算下去。

下面我们尝试设计基于 DM 方案的鉴权程序来计算 18 比特 Hash 值。提出的鉴权系统如图 8.17 所示。

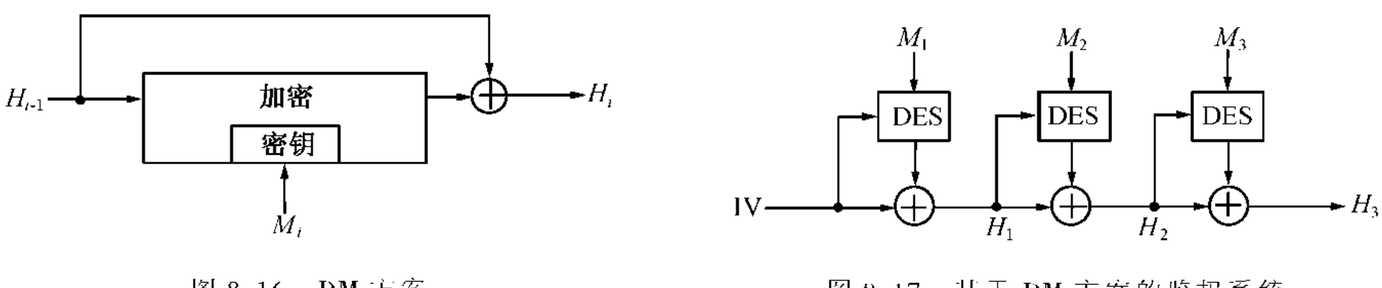


图 8.16　DM 方案　　　　图 8.17　基于 DM 方案的鉴权系统

首先将要进行 Hash 变换的消息等分成一系列 m 比特的数据块，为了等分消息需要进行适当的填充，n 比特的消息加上 k 比特的填充就是把 n 比特值映射成$(n+k)$比特的值，然后将$(n+k)$比特的消息分成一系列等长的 m 比特块。

基于图 8.17 提出的鉴权过程描述如下：

1. 填充 40 比特将 152 比特消息变成 192 比特的消息；
2. 将 192 比特已填充的消息三等分成 64 比特的消息块，每个块将用做 DES 密钥；
3. 随机选择 64 比特作为初始向量 IV(或 H_0)，并把它用做 DES 算法的明文输入；
4. 与 H_0 异或计算出输出密文 H_1，并且把 H_1 作为对第二个消息块进行 Hash 运算的输入；
5. 下一个 64 比特的消息块 M_2 成为新的 DES 密钥；
6. H_2 成为密文输出，$H_2 = E_{M_2}(H_1) \oplus H_1$，并成为第三个消息块进行运算的输入；
7. H_3 变成 64 比特的值，$H_3 = E_{M_3}(H_2) \oplus H_2$，从 H_3 可以计算出 18 比特的消息摘要(AUTHR 数据)。

例 8.7　将192比特已填充的消息分成三个 64 比特的消息块，即 M_1，M_2 和 M_3，它们将用做 DES 密钥。

设

初始向量(IV)：

$$H_0(\text{IV}) = 67542301\text{efcdab}89$$

DES 密钥数据：

$$M_1 = 7\text{a}138\text{b}2524\text{af}17\text{c}3$$

$$M_2 = 17\text{b}439\text{a}12\text{f}51\text{c}5\text{a}8$$

$$M_3 = 51\text{cb}360000000000$$

I. 第一个 DES 运算(见图 8.17 所示)

1. 密钥方案

对密钥数据 M_1 应用表 8.10(PC-1)，PC-1 将输出 a481394(左半)和 e778253(右半)，根据表

8.12，这两半同时向左移一个或两个位置后串联。计算出的从第 1 轮到第 16 轮的 48 比特的密钥数据显示如下：

K_1 = 058c457c2c45 K_2 = d805187690e5

K_3 = 4368a26ecode K_4 = 44822dc2aceb

K_5 = b8a5a061f48b K_6 = 8398062ebf19

K_7 = d0060bae146b K_8 = 282ae27f5472

K_9 = 619214cadb66 K_{10} = b074284dc82a

K_{11} = 0498c614cff8 K_{12} = c00750c47c5c

K_{13} = 326072dd9c51 K_{14} = 44da15e993fc

K_{15} = ac4520cbe738 K_{16} = 07b14291dfab

2. 加密

用密钥 M_1 加密 H_0(IV)，如下：

要加密的 64 比特 H_0 是

$$H_0 = 67452301\text{efcdab}89$$

使用表 8.6 进行初始置换后的数据为

$$\text{IP} = 330033\text{fff}055\text{f}055$$

其中 L_0 = 330033ff，R_0 = f055f055

利用表 8.7 将初始置换后的数据 R_0 扩展到 48 比特

$$E(R_0) = \text{fa02abfa02ab}$$

将扩展后的数据和密钥数据 K_1 进行异或运算，输出为：

$$\begin{aligned}\Gamma_1 &= E(R_0)\oplus K_1(\text{fa02abfa02ab})\oplus(058\text{c}457\text{c}2\text{c}45)\\ &= \text{ff}8\text{eee}862\text{eee}\end{aligned}$$

将这个序列输入到(S-box)$_1$ 中，从表 8.8 得到(S-box)$_1$ 的输出为

$$\Omega_1 = \text{d}95\text{dbe}22$$

使用表 8.9 将(S-box)$_1$ 的输出进行置换后得到

$$P(\Omega_1) = \text{b}5\text{abd}4\text{ca}$$

最后，可以通过下面的公式得到 32 比特的 R_1

$$\begin{aligned}R_1 &= P(\Omega_1)\oplus L_0\\ &= (\text{b}5\text{abd}4\text{ca})\oplus(330033\text{ff})\\ &= 86\text{abc}735\end{aligned}$$

至此，已经完成了 H_0 的第 1 轮加密计算。

一直要计算到第 16 轮的确是非常冗长的，但是每一轮的计算过程正好是完全相同的，因此，将 16 轮的所有结果总结如下。

考虑 16 轮的 DES 加密使用(H_0，M_1)：

H_0 = 67452301efcdab89

M_1 = 7a138b2524af17c3

第 1 轮 **第 2 轮**

$K_1 = 058c457c2c45$

$L_0 = 330033ff, R_0 = f055f055$

$E(R_0) = fa02abfa02ab$

$\Gamma_1 = E(R_0) \oplus K_1$

$= ff8eee862eee$

$\Omega_1 = d95dbe22$

$P(\Omega_1) = b5abd4ca$

$R_1 = P(\Omega_1) \oplus L_0$

$= 86abe735$

$K_2 = d805187690e5$

$R_1 = 86abe735$

$E(R_1) = c0d557f0e9ab$

$\Gamma_2 = E(R_1) + K_2$

$= 18d04f86794e$

$\Omega_2 = 18d3bcd1$

$P(\Omega_2) = b759098b$

$R_2 = P(\Omega_2) \oplus L_1 = P(\Omega_2) \oplus R_0$

$= 470cf9de$

第 **3** 轮

$K_3 = 4368a26ee0de$

$R_2 = f470cf9de$

$E(R_2) = 20e8597f3efc$

$\Gamma_3 = E(R_2) \oplus K_3$

$= 6380fb11de22$

$\Omega_3 = 5977430b$

$P(\Omega_3) = 8c6ff806$

$R_3 = P(\Omega_3) \oplus R_1$

$= 0ac41f33$

第 **4** 轮

$K_4 = 44822dc2aceb$

$R_3 = 0ac41f33$

$E(R_3) = 8556080fe9a6$

$\Gamma_4 = E(R_3) \oplus K_4$

$= c1d425cd454d$

$\Omega_4 = fb10f357$

$P(\Omega_4) = 67beeaa2$

$R_4 = P(\Omega_4) \oplus R_2$

$= 20b2137c$

第 **5** 轮

$K_5 = b8a5a061f48b$

$R_4 = 20b2137c$

$E(R_4) = 1015a4a6bf8$

$\Gamma_5 = E(R_4) \oplus K_5$

$= a8b0046b9f73$

$\Omega_5 = 62ae063c$

$P(\Omega_5) = 4a60976c$

$R_5 = P(\Omega_5) \oplus R_3$

$= 40a4885f$

第 **6** 轮

$K_6 = 8398062ebf19$

$R_5 = 40a4885f$

$E(R_5) = a015094502fe$

$\Gamma_6 = E(R_5) \oplus K_6$

$= 238d0f6bbde7$

$\Omega_6 = 292300f7$

$P(\Omega_6) = 825a4e25$

$R_6 = P(\Omega_6) \oplus R_4$

$= a2e85d59$

第 **7** 轮

$K_7 = d0060bae146b$

$R_6 = a2e85d59$

$E(R_6) = d057502faaf3$

$\Gamma_7 = E(R_6) \oplus K_7$

$= 00515b81be98$

$\Omega_7 = e40a4b55$

$P(\Omega_7) = 12f4aa70$

$R_7 = P(\Omega_7) \oplus R_5$

$= 05250222f$

第 **8** 轮

$K_8 = 282ae27f5472$

$R_7 = 5250222f$

$E(R_7) = aa42a010415e$

$\Gamma_8 = E(R_7) \oplus K_8$

$= 8268426f152c$

$\Omega_8 = 4b1d9b9e$

$P(\Omega_8) = ff2af061$

$R_8 = P(\Omega_8) \oplus R_6$

$= 5dc2ad38$

第 9 轮

$K_9 = 619214cadb66$

$R_8 = 5dc2ad38$

$E(R_8) = 2fbe0555a9f0$

$\Gamma_9 = E(R_8) \oplus K_9$

$= 4e2c119f7296$

$\Omega_9 = 6eb4770e$

$P(\Omega_9) = 6c2eb3bc$

$R_9 = P(\Omega_9) \oplus R_7$

$= 3e7e9193$

第 10 轮

$K_{10} = b074284dc82a$

$R_9 = 3e7e9193$

$E(R_9) = 9fc3fd4a3ca6$

$\Gamma_{10} = E(R_9) \oplus K_{10}$

$= 2fb7d507f48c$

$\Omega_{10} = 12512edcb$

$P(\Omega_{10}) = 1d566a99$

$R_{10} = P(\Omega_{10}) \oplus R_8$

$= 4094c7a1$

第 11 轮

$K_{11} = 0498c614cff8$

$R_{10} = 4094c7a1$

$E(R_{10}) = a014a960fd02$

$\Gamma_{11} = E(R_{10}) \oplus K_{11}$

$= a48c6f7432fa$

$\Omega_{11} = 46488f93$

$P(\Omega_{11}) = 5323a859$

$R_{11} = P(\Omega_{11}) \oplus R_9$

$= 6d5d39ca$

第 12 轮

$K_{12} = c00750c47c5c$

$R_{11} = 6d5d39ca$

$E(R_{11}) = 35aafa9f3e54$

$\Gamma_{12} = E(R_{11}) \oplus K_{12}$

$= f5adaa5b4208$

$\Omega_{12} = 60cbf4f6$

$P(\Omega_{12}) = a35787e9$

$R_{12} = P(\Omega_{12}) \oplus R_{10}$

$= e3c34048$

第 13 轮

$K_{12} = 326072dd9c51$

$R_{13} = e3c34048$

$E(R_{12}) = 707e06a00251$

$\Gamma_{13} = E(R_{12}) \oplus K_{13}$

$= 421e747d9e00$

$\Omega_{13} = 3db3600d$

$P(\Omega_{13}) = 8c4c4bb6$

$R_{13} = P(\Omega_{13}) \oplus R_{11}$

$= e111727c$

第 14 轮

$K_{14} = 44da15e993fc$

$R_{13} = e111727c$

$E(R_{13}) = 7028a2ba43f9$

$\Gamma_{14} = E(R_{13}) \oplus K_{14}$

$= 34f2b753d005$

$\Omega_{14} = de3b384d$

$P(\Omega_{14}) = fcd888f6$

$R_{14} = P(\Omega_{14}) \oplus R_{12}$

$= 1f1bc8be$

第 15 轮

$K_{15} = ac4520cbe738$

$R_{14} = 1f1bc8be$

$E(R_{14}) = fe8f7e515fc$

$\Gamma_{15} = E(R_{14}) \oplus K_{15}$

$= a3add72ef2c4$

$\Omega_{15} = d33c7a98$

$P(\Omega_{15}) = 7ea4d0c7$

第 16 轮

$K_{16} = 07b14291dfab$

$R_{15} = 9fb5a2bb$

$E(R_{15}) = effdabd055f7$

$\Gamma_{16} = E(R_{15}) \oplus K_{16}$

$= c84ce9418a5c$

$\Omega_{16} = c8fa8e1c$

$P(\Omega_{16}) = 1fe9816c$

$R_{15} = P(\Omega_{15}) \oplus R_{13}$
$= 9fb5a2bb$

$R_{16} = P(\Omega_{16}) \oplus R_{14}$
$= 00f249d2$

预输出块（$R_{16} \parallel R_{15}$）是 R_{16} 和 R_{15}（$= L_{16}$）的串联，使用表 8.13 对预输出块（$R_{16} \parallel R_{15}$）进行反置换 IP^{-1} 得到第一个 DES 算法的输出，$E_{M_1}(H_0)$。

$$R_{16} \parallel R_{15} = 00f249d29fb5a2bb$$

这样，从 IP^{-1} 得到的数据为

$$IP^{-1} = a69ba086b33a15bb$$

IP^{-1} 与 H_0 异或得到第二个 DES 算法的输入 H_1

$$H_1 = IP^{-1} \oplus H_0$$
$$= (a69ba086b33a15bb) \oplus (67452301efcdab89)$$
$$= c1de83875cf7be32$$

表 8.13　初始置换的反置换，IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

II. 第二个 DES 运算（见图 8.17）

密钥 M_2：17b439a12f51c5a8

输入 H_1：c1de83875cf7be32

1. 密钥方案

对密钥数据 M_2 应用表 8.10（PC-1），PC-1 的输出为 ca609e2（左半）和 1153947（右半），这两半根据表 8.12 移位，产生从第 1 轮到第 16 轮的 48 比特的密钥数据，显示如下：

$K_1 = 0975a4d9e052$　　$K_2 = 07a44cc18bad$
$K_3 = d005dd21e708$　　$K_4 = 4a4cb2161f99$
$K_5 = 55c221b83446$　　$K_6 = fca9285b1035$
$K_7 = 8399a6a8c3a6$　　$K_8 = 82a60b0768a8$
$K_9 = b822c7106fc3$　　$K_{10} = 691e16643815$
$K_{11} = 315628ba8051$　　$K_{12} = 64b8c8e700be$
$K_{13} = c011f483e742$　　$K_{14} = 12e472411b8b$
$K_{15} = 94ca553ca700$　　$K_{16} = ec4d1216107d$

2. 加密

用密钥 M_2 加密输入 H_1，如下：

使用表 8.6 进行初始置换后的数据为

$$\text{IP} = 33\text{f}27\text{a}2\text{d}6\text{fc}052\text{ee}$$

其中 $L_0 = 33\text{f}27\text{a}2\text{d}$，$R_0 = 6\text{fc}052\text{ee}$

利用表 8.7 将 32 比特的数据 R_0 扩展到 48 比特的 $E(R_0)$

$$E(R_0) = 35\text{ff}002\text{a}575\text{c}$$

将 $E(R_0)$和密钥数据 K_1 进行异或运算，输出：

$$\Gamma_1 = E(R_0) \oplus K_1 = (35\text{ff}002\text{a}575\text{c}) \oplus (0975\text{a}4\text{d}9\text{e}052)$$
$$= 3\text{c}8\text{aa}4\text{f}3\text{b}70\text{e}$$

将这个序列输入到(S-box)$_1$ 中，使用表 8.8 得到(S-box)$_1$ 的输出 Ω_1

$$\Omega_1 = 16\text{f}90061$$

使用表 8.9 得到 Ω_1 置换后的数据为

$$P(\Omega_1) = \text{c}4110\text{d}56$$

这样，可以通过下面的公式得到 32 比特的 R_1

$$R_1 = P(\Omega_1) \oplus L_0$$
$$= (\text{c}4110\text{d}56) \oplus (33\text{f}27\text{a}2\text{d})$$
$$= \text{f}7\text{e}3777\text{b}$$

这就是 H_1 的第 1 轮加密结果。

使用(H_1，M_2)进行加密的 16 轮的加密过程总结如下：

第 1 轮

$K_1 = 0975\text{a}4\text{d}9\text{e}052$

$L_0 = 33\text{f}27\text{a}2\text{d}$，$R_0 = 6\text{fe}052\text{ee}$

$E(R_0) = 35\text{ff}002\text{a}575\text{c}$

$\Gamma_1 = E(R_0) \oplus K_1$

$= 3\text{c}8\text{aa}4\text{f}3\text{b}70\text{e}$

$\Omega_1 = 16\text{f}90061$

$P(\Omega_1) = \text{c}4110\text{d}56$

$R_1 = P(\Omega_1) \oplus L_0$

$= (\text{c}4110\text{d}56) \oplus (33\text{f}27\text{a}2\text{d})$

$= \text{f}7\text{e}3777\text{b}$

第 2 轮

$K_2 = 07\text{a}44\text{cc}18\text{bad}$

$R_1 = \text{f}7\text{e}3777\text{b}$

$E(R_1) = \text{faff}06\text{baebf}7$

$\Gamma_2 = E(R_1) \oplus K_2$

$= \text{fd}5\text{b}4\text{a}7\text{b}605\text{a}$

$\Omega_2 = \text{d}1869\text{ad}0$

$P(\Omega_2) = 33\text{f}0\text{d}103$

$R_2 = P(\Omega_2) \oplus R_0$

$= 5\text{c}1083\text{ed}$

第 3 轮

$K_3 = \text{d}005\text{dd}21\text{e}708$

$R_2 = 75\text{c}1083\text{cd}$

$E(R_2) = \text{af}80\text{a}1407\text{f}5\text{a}$

$\Gamma_3 = E(R_2) \oplus K_3$

$= 7\text{f}857\text{c}619852$

$\Omega_3 = 8958\text{d}069$

$P(\Omega_3) = 2\text{d}9\text{d}4\text{c}40$

$R_3 = P(\Omega_3) \oplus R_1$

$= \text{da}7\text{e}3\text{b}3\text{b}$

第 4 轮

$K_4 = 4\text{a}4\text{cb}2161\text{f}99$

$R_3 = \text{da}7\text{e}3\text{b}3\text{b}$

$E(R_3) = \text{ef}43\text{fc}1\text{f}69\text{f}7$

$\Gamma_4 = E(R_3) \oplus K_4$

$= \text{a}50\text{f}4\text{e}09766\text{e}$

$\Omega_4 = \text{d}492\text{ace}22$

$P(\Omega_4) = 116\text{ec}44\text{c}$

$R_4 = P(\Omega_4) \oplus R_2$

$= 4\text{d}7\text{e}47\text{a}1$

第 **5** 轮

$K_5 = 55c221b83446$

$R_4 = 4d7e47a1$

$E(R_4) = a5abfc20fd02$

$\Gamma_5 = E(R_4) \oplus K_5$

$= f069dd98c944$

$\Omega_5 = 5e0eb6d8$

$P(\Omega_5) = 6b7890db$

$R_5 = P(\Omega_5) \oplus R_3$

$= b106abe0$

第 **6** 轮

$K_6 = fca9285b1035$

$R_5 = b106abe0$

$E(R_5) = 5a280d557f01$

$\Gamma_6 = E(R_5) \oplus K_6$

$= a681250e6f34$

$\Omega_6 = 4a90b59a$

$P(\Omega_6) = 6f0aa189$

$R_6 = P(\Omega_6) \oplus R_4$

$= 2274e628$

第 **7** 轮

$K_7 = 8399a6a8c3a6$

$R_6 = 2274e628$

$E(R_6) = 1043a970c150$

$\Gamma_7 = E(R_6) \oplus K_7$

$= 93da0fd802f6$

$\Omega_7 = e4835c9d$

$P(\Omega_7) = bac48b39$

$R_7 = P(\Omega_7) \oplus R_5$

$= bc220d9$

第 **8** 轮

$K_8 = 82a60b0768a8$

$R_7 = bc220d9$

$E(R_7) = 857e041016f2$

$\Gamma_8 = E(R_7) \oplus K_8$

$= 07d80f177e5a$

$\Omega_8 = 04d327e0$

$P(\Omega_8) = 84712599$

$R_8 = P(\Omega_8) \oplus R_6$

$= a605c3b1$

第 **9** 轮

$K_9 = b822c7106fc3$

$R_8 = a605c3b1$

$E(R_8) = d0c00be07da3$

$\Gamma_9 = E(R_8) \oplus K_9$

$= 68e2eef01260$

$\Omega_9 = 94490a47$

$P(\Omega_9) = 90b30872$

$R_9 = P(\Omega_9) \oplus R_7$

$= 9b7128ab$

第 **10** 轮

$K_{10} = 691e16643815$

$R_9 = 9b7128ab$

$E(R_9) = cf6ba2951557$

$\Gamma_{10} = E(R_9) \oplus K_{10}$

$= a675b4f12d42$

$\Omega_{10} = 41730d02$

$P(\Omega_{10}) = 9443e00c$

$R_{10} = P(\Omega_{10}) \oplus R_8$

$= 324623bd$

第 **11** 轮

$K_{11} = 315628ba8051$

$R_{10} = 324623bd$

$E(R_{10}) = 9a420c107dfa$

$\Gamma_{11} = E(R_{10}) \oplus K_{11}$

$= ab1424afdab$

$\Omega_{11} = 6b19da8a$

$P(\Omega_{11}) = fd2ec241$

第 **12** 轮

$K_{12} = 64b8c8e700be$

$R_{11} = 665feaea$

$E(R_{11}) = 30c2fff55754$

$\Gamma_{12} = E(R_{11}) \oplus K_{12}$

$= 547a371257ea$

$\Omega_{12} = c78b426c$

$P(\Omega_{12}) = c8f4c570$

$R_{11} = P(\Omega_{11}) \oplus R_9$
$= 665feaea$

$R_{12} = P(\Omega_{12}) \oplus R_{10}$
$= fab2e6cd$

第 **13** 轮

$K_{13} = c011f483e742$

$R_{12} = fab2e6cd$

$E(R_{12}) = 30c2fff55754$

$\Gamma_{13} = E(R_{12}) \oplus K_{13}$
$= 3f4451f33119$

$\Omega_{13} = 1c240e20$

$P(\Omega_{13}) = 1028141e$

$R_{13} = P(\Omega_{13}) \oplus R_{11}$
$= 7677fef4$

第 **14** 轮

$K_{14} = 12e472411b8b$

$R_{13} = 7677fef4$

$E(R_{13}) = 3ac3afffd7a8$

$\Gamma_{14} = E(R_{13}) \oplus K_{14}$
$= 2827ddbecc23$

$\Omega_{14} = f11edca1$

$P(\Omega_{14}) = 35c4de4b$

$R_{14} = P(\Omega_{14}) \oplus R_{12}$
$= cf763886$

第 **15** 轮

$K_{15} = 94ca553ca700$

$R_{14} = cf763886$

$E(R_{14}) = 65ebac1f140d$

$\Gamma_{15} = E(R_{14}) \oplus K_{15}$
$= f121f923b30d$

$\Omega_{15} = 579c7087$

$P(\Omega_{15}) = 6406d9f3$

$R_{15} = P(\Omega_{15}) \oplus R_{13}$
$= 12712707$

第 **16** 轮

$K_{16} = ec4d1216107d$

$R_{15} = 12712707$

$E(R_{15}) = 8a43a290e80e$

$\Gamma_{16} = E(R_{15}) \oplus K_{16}$
$= 660eb086f873$

$\Omega_{16} = 90afba6c$

$P(\Omega_{16}) = b9f015e6$

$R_{16} = P(\Omega_{16}) \oplus R_{14}$
$= 76862d60$

预输出块是 R_{16} 和 $R_{15}(=L_{16})$ 的串联，即 $R_{16} \parallel R_{15}$，使用表 8.13 对预输出块 $(R_{16} \parallel R_{15})$ 进行反置换 IP^{-1} 得到第二个 DES 算法的输出，$E_{M_2}(H_1)$。

$$R_{16} \parallel R_{15} = 76862d6012712707$$

这样，从 IP^{-1} 得到的数据为

$$IP^{-1} = 2eda5e04e06d6110$$

IP^{-1} 与 H_1 异或得到第二个 DES 算法的输入 H_2

$$\begin{aligned} H_2 &= IP^{-1} \oplus H_1 \\ &= (2eda5e04e06d6110) \oplus (c1de83875cf7be32) \\ &= ef04dd83bc9adf22 \end{aligned}$$

III. 第三个 DES 运算（见图 8.17 所示）

密钥 M_3：51cb360000000000

输入 H_2：ef04dd83bc9adf22

1. 密钥方案

对密钥数据 M_3 应用表 8.10（PC-1），PC-1 的输出为 203040（左半）和 604025（右半），这两半根据表 8.12 移位，产生从第 1 轮到第 16 轮的 48 比特的密钥数据，显示如下：

K_1 = 80e000209341	K_2 = 8400066a0008
K_3 = 000702128422	K_4 = 22020000500e
K_5 = 601001080c00	K_6 = 0810200410a0
K_7 = 0180400c6050	K_8 = 800058c00861
K_9 = 00409261c000	K_{10} = 044200028b18
K_{11} = 34010080040a	K_{12} = 021900111410
K_{13} = 020001881304	K_{14} = 0820410d0020
K_{15} = 0900141043e0	K_{16} = 014408406804

2. 加密

用密钥 M_3 加密输入 H_2，如下：

使用表 8.6 进行初始置换后的数据为

$$IP = 4574574d7d9175e9$$

利用表 8.7 将 32 比特的数据 R_0 扩展到 48 比特的 $E(R_0)$

$$E(R_0) = bfbca2babf52$$

将 $E(R_0)$和密钥数据 K_1 进行异或运算，输出：

$$\Gamma_1 = E(R_0) \oplus K_1 = (bfbca2babf52) \oplus (80e000209341)$$
$$= 3f5ca29a2c13$$

将这个数据序列 Γ_1 输入到$(S\text{-box})_1$ 中，使用表 8.8 得到$(S\text{-box})_1$ 的输出 Ω_1 为

$$\Omega_1 = 1716ea5$$

利用表 8.9 得到 Ω_1 置换后的数据为

$$P(\Omega_1) = 75605cbb$$

这样，可以通过下面的公式得到 32 比特的 R_1

$$R_1 = P(\Omega_1) \oplus L_0$$
$$= (75605cbb) \oplus (4574574d)$$
$$= 30140bf6$$

这就是 H_2 的第 1 轮加密结果。使用(H_2, M_3)进行加密的 16 轮的加密过程总结如下：

16 轮 DES 加密开始时要用到下列数据：

$$H_2 = ef04dd83bc9adf22$$
$$M_3 = 51cb36000000000$$

第 1 轮

K_1 = 80e000209341

L_0 = 4574574d, R_0 = 7d9175e9

$E(R_0)$ = bfbca2babf52

$\Gamma_1 = E(R_0) \oplus K_1$

= 3f5ca29a2c13

Ω_1 = 1716bea5

$P(\Omega_1)$ = 75605cbb

$R_1 = P(\Omega_1) \oplus L_0$

第 2 轮

K_2 = 8400066a0008

R_1 = 30140bf6

$E(R_1)$ = 1a00a8057fac

$\Gamma_2 = E(R_1) \oplus K_2$

= 9e00ae6f7fa4

Ω_2 = 200d9724

$P(\Omega_2)$ = a1203668

$R_2 = P(\Omega_2) \oplus R_0$

$=30140bf6$

第 **3** 轮

$K_3=000702128422$

$R_2=dcb14381$

$E(R_2)=ef95a2a07c03$

$\Gamma_3=E(R_2)\oplus K_3$

$=ef92a0b2f821$

$\Omega_3=003a7a12$

$P(\Omega_3)=366600c4$

$R_3=P(\Omega_3)\oplus R_1$

$=06720b32$

第 **5** 轮

$K_5=601001080c00$

$R_4=57ac2074$

$E(R_4)=2afd581003a8$

$\Gamma_5=E(R_4)\oplus K_5$

$=4aed59180fa8$

$\Omega_5=a1e11c29$

$P(\Omega_5)=b8814f0c$

$R_5=P(\Omega_5)\oplus R_3$

$=bef3443e$

第 **7** 轮

$K_7=0180400c6050$

$R_6=bd4c8d7d$

$E(R_6)=dfaa5945abfb$

$\Gamma_7=E(R_6)\oplus K_7$

$=de2a1949cbab$

$\Omega_7=ee8155ea$

$P(\Omega_7)=e89ea719$

$R_7=P(\Omega_7)\oplus R_5$

$=566de327$

第 **9** 轮

$K_9=00409261c000$

$R_8=82e56282$

$E(R_8)=40750ab05405$

$\Gamma_9=E(R_8)\oplus K_9$

$=401798d19405$

$\Omega_9=338bc03d$

$P(\Omega_9)=cb444f62$

$R_9=P(\Omega_9)\oplus R_7$

$=dcb14381$

第 **4** 轮

$K_4=22020000500e$

$R_3=06720b32$

$E(R_3)=00c3a40569a4$

$\Gamma_4=E(R_3)\oplus K_4$

$=22c1a40539aa$

$\Omega_4=2de9e1dc$

$P(\Omega_4)=8b1d63f5$

$R_4=P(\Omega_4)\oplus R_2$

$=57ac2074$

第 **6** 轮

$K_6=0810200410a0$

$R_5=bef3443e$

$E(R_5)=5fd7a6a081fd$

$\Gamma_6=E(R_5)\oplus K_6$

$=57c786a4915d$

$\Omega_6=c28317b9$

$P(\Omega_6)=eae0ad09$

$R_6=P(\Omega_6)\oplus R_4$

$=bd4c8d7d$

第 **8** 轮

$K_8=800058c00861$

$R_7=566de327$

$E(R_7)=aac35bf0690e$

$\Gamma_8=E(R_7)\oplus K_8$

$=2ac30330616f$

$\Omega_8=fdf8bfbd$

$P(\Omega_8)=3fa9efff$

$R_8=P(\Omega_8)\oplus R_6$

$=82e56282$

第 **10** 轮

$K_{10}=044200028b18$

$R_9=9d29ac45$

$E(R_9)=cfa953d5820b$

$\Gamma_{10}=E(R_9)\oplus K_{10}$

$=cbeb53d70913$

$\Omega_{10}=cf8707b5$

$P(\Omega_{10})=c2e8fd39$

$R_{10}=P(\Omega_{10})\oplus R_8$

$=9d29ac45$

第 11 轮

$K_{11}=34010080040a$

$R_{10}=400d9fbb$

$E(R_{10})=a0005bcffdf6$

$\Gamma_{11}=E(R_{10})\oplus K_{11}$

$=94015b4ff9fc$

$\Omega_{11}=8f0a0d85$

$P(\Omega_{11})=50c86879$

$R_{11}=P(\Omega_{11})\oplus R_{9}$

$=cde1c43c$

$=400d9fbb$

第 12 轮

$K_{12}=021900111410$

$R_{11}=cde1c43c$

$E(R_{11})=65bf03e081f9$

$\Gamma_{12}=E(R_{11})\oplus K_{12}$

$=67a603f195e9$

$\Omega_{12}=93b800c4$

$P(\Omega_{12})=44904167$

$R_{12}=P(\Omega_{12})\oplus R_{10}$

$=049ddedc$

第 13 轮

$K_{13}=020001881304$

$R_{12}=049ddedc$

$E(R_{12})=0094fbefd6f8$

$\Gamma_{13}=E(R_{12})\oplus K_{13}$

$=0294fa67c5fc$

$\Omega_{13}=e3823bc5$

$P(\Omega_{13})=70f0eba1$

$R_{13}=P(\Omega_{13})\oplus R_{11}$

$=bd112f9d$

第 14 轮

$K_{14}=0820410d0020$

$R_{13}=bd112f9d$

$E(R_{13})=dfa8a295fcfb$

$\Gamma_{14}=E(R_{13})\oplus K_{14}$

$=d788e398fcdb$

$\Omega_{14}=39afb55e$

$P(\Omega_{14})=ab5a73ee$

$R_{14}=P(\Omega_{14})\oplus R_{12}$

$=afc7ad32$

第 15 轮

$K_{15}=0900141043e0$

$R_{14}=afc7ad32$

$E(R_{14})=55fe0fd5a9a5$

$\Gamma_{15}=E(R_{14})\oplus K_{15}$

$=5cfe1bc5ea45$

$\Omega_{15}=be5a6b1d$

$P(\Omega_{15})=5eed2af2$

$R_{15}=P(\Omega_{15})\oplus R_{13}$

$=e3fc056f$

第 16 轮

$K_{16}=014408406804$

$R_{15}=e3fc056f$

$E(R_{15})=f07ff800ab5f$

$\Gamma_{16}=E(R_{15})\oplus K_{16}$

$=f13bf040c35b$

$\Omega_{16}=507f861e$

$P(\Omega_{16})=8f63906e$

$R_{16}=P(\Omega_{16})\oplus R_{14}$

$=20a43d5c$

这样，预输出块是

$$R_{16}\parallel R_{15}=20a43d5ce3fc056f$$

使用表 8.13，从 IP^{-1} 得到的数据为

$$IP^{-1}=8e823f2725f6a3b0$$

IP^{-1} 与 H_2 异或得到第三个 DES 算法的输出如下：

$$\begin{aligned}H_3&=IP^{-1}\oplus H_2\\&=(8e823f2725f6a3b0)\oplus(ef04dd83bc9adf22)\\&=6186f2a4996c7e92\end{aligned}$$

先将这个十六进制的数转换成二进制数，然后在这个二进制序列两端分别去掉 5 个比特，剩余的比特序列如下：

001100001101111001010100100100110010110110001111110100

将抽取规则应用于这个序列，每 3 个比特取 1 个比特，得到需要的 18 比特鉴权数据（AUTHR）为：

AUTHR = 101111000000001100

8.11.5 18 位 AUTHR 计算的另一种方法

图 8.18 说明了另一种用于计算 18 比特 AUTHR 的方法，这里我们要考虑两种情况：(1)152 比特的消息加上 24 比特的填充构成的 176 比特已填充消息；(2)添加 40 比特的填充构成 192 比特的已填充消息。

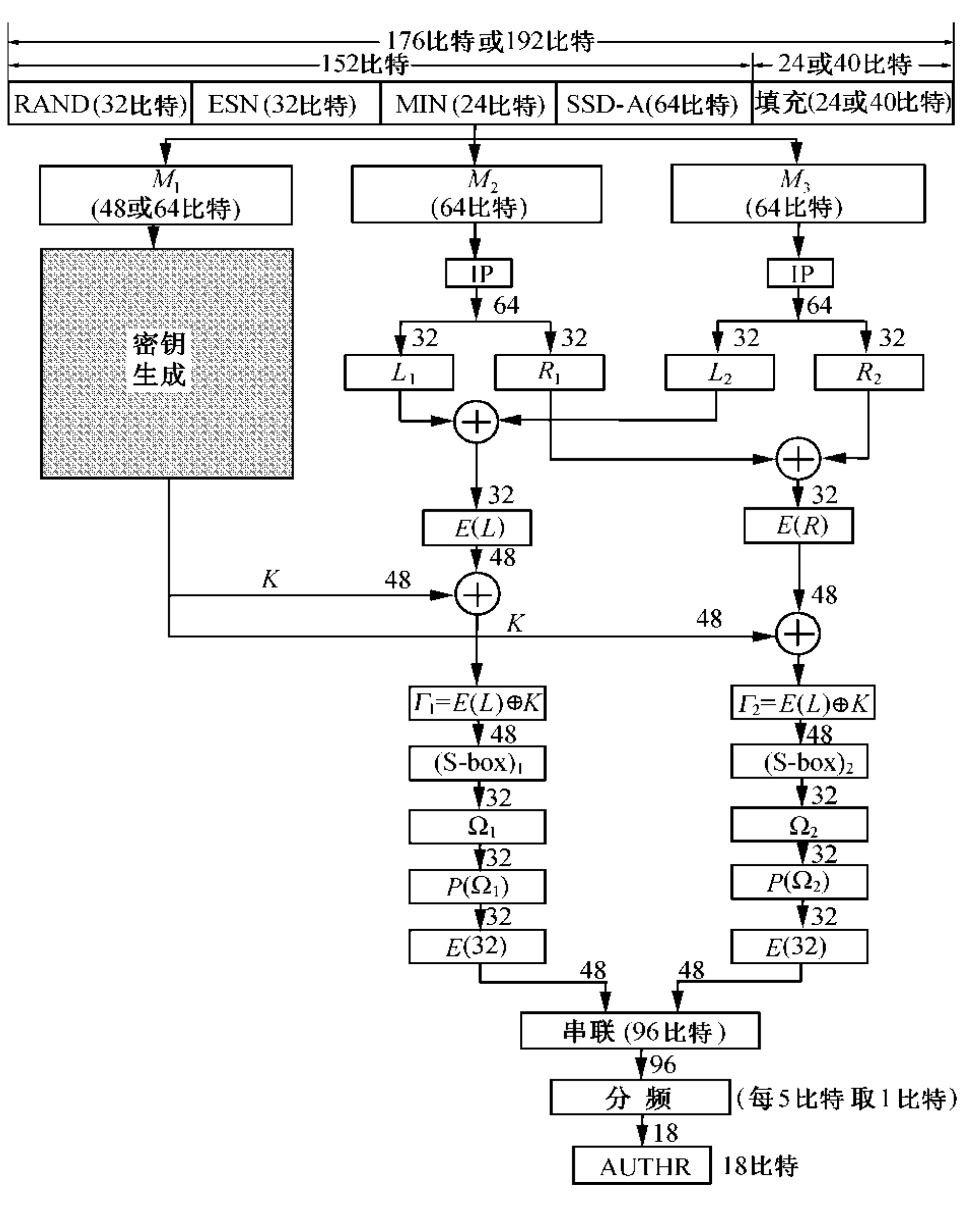


图 8.18 18 比特 AUTHR 计算的另一种方法

通过下面的例子说明 AUTHR 的计算过程。

例 8.8 假设输入参数的设置如下：

M_1(48 比特，密钥输入) = 16c27a415f39

M_2(64 比特) = 17b439a12f51c5a8

M_3(64 比特) = 51cb36af4300000

如同 8.11.1 节中的密钥生成技术(I),可以通过行列置换的方法得到 48 比特的密钥,如下所示:

$$K = 01959d7857e4$$

使用表 8.6 计算出 M_2 经过初始置换后为

$$(IP)_1 = 6027537dca9e9411$$

其中 L_1 = 6027537d,R_1 = ca9e9411

同样,M_3 经过初始置换后为

$$(IP)_2 = 13050c1b0a0c0a1e$$

其中 L_2 = 13050c1b,R_2 = 0a0c0a1e

L_1 和 L_2 异或,得到

$$L = L_1 \oplus L_2 = 73225f66$$

R_1 和 R_2 异或,得到

$$R = R_1 \oplus R_2 = c0929e0f$$

利用表 8.7 将 32 比特的 L 扩展到 48 比特

$$E(L) = 3a69042feb0c$$

$E(L)$ 与 48 比特 K 模 2 加,所得结果作为$(\text{S-box})_1$ 的输入:

$$\begin{aligned}\Gamma_1 &= E(L) \oplus K \\ &= (3a69042feb0c) \oplus (01959d7857e4) \\ &= 3dfc9957bce8\end{aligned}$$

使用表 8.8 得到$(\text{S-box})_1$ 的 32 比特输出 Ω_1 为

$$\Omega_1 = 8911f059$$

根据表 8.9 得到 Ω_1 置换后的数据为

$$P(\Omega_1) = af9c4880$$

根据表 8.7 将 32 比特的 $P(\Omega_1)$ 扩展到 48 比特,得到

$$E[P(\Omega_1)] = 55fcf8251401$$

接着,使用表 8.7 将 R 扩展到 48 比特,得到

$$E(R) = e014a54fc05f$$

$E(R)$ 与 48 比特 K 模 2 加,所得结果 Γ_2 作为$(\text{S-box})_2$ 的输入:

$$\begin{aligned}\Gamma_2 &= E(R) \oplus K \\ &= (e014a54fc05f) \oplus (01959d7857e4) \\ &= e181383797bb\end{aligned}$$

使用表 8.8 得到$(\text{S-box})_2$ 的 32 比特输出 Ω_2 为

$$\Omega_2 = 3c95d615$$

根据表 8.9 得到 Ω_2 置换后的数据为

$$P(\Omega_2) = a72c1b3a$$

使用表 8.7 将 32 比特的 $P(\Omega_2)$ 扩展到 48 比特,得到

$$E[P(\Omega_2)] = 50e9580f69f5$$

最后,串联 $E[P(\Omega_1)]$ 和 $E[P(\Omega_2)]$ 得到

55fcf825140150e9580f69f5(96 比特)

先将这个十六进制的数转换成二进制数，然后在这个二进制序列两端分别去掉 12 个比特，剩余比特序列如下：

1100 1111 1000 0010 0101 0001 0100 0000 0001

0101 0000 1110 1001 0101 1000 0000 1111 0110

将抽取规则应用于这个序列，每 4 个比特取 1 个比特，得到 18 比特鉴权数据如下：

AUTHR = 010011001100110010

例 8.9 考虑通过添加40比特的填充构成 192 比特的已填充消息这种情况。输入参数的设置如下：

M_1(密钥输入，64 比特) = 7a138b2524af17c3

M_2(输入数据，64 比特) = 17b439a12f51c5a8

M_3(输入数据，64 比特) = 51cb360000000000

利用 8.11.2 节中密钥生成技术(II)(见图 8.13 所示)，计算出 48 比特的密钥数据

K = 058c4517a7a2

这就是 PC-2 的结果

根据表 8.6 计算出 M_2 经过初始置换后为

$(IP)_1$ = 6027537dca9e9411

其中 L_1 = 6027537d，R_1 = ca9e9411

同样，M_3 经过初始置换后为

$(IP)_2$ = 0305040302040206

其中 L_2 = 03050403，R_2 = 02040206

分别将 L_1 和 L_2 异或，R_1 和 R_2 异或，得到

$$L = L_1 \oplus L_2 = 6322577e$$

和

$$R = R_1 \oplus R_2 = c89a9617$$

利用表 8.7 将 L 和 R 扩展到 48 比特，得到

$E(L)$ = 3069042aebfc

$E(R)$ = e514f54ac0af

$E(L)$与 48 比特 K 模 2 加，所得结果作为$(S\text{-box})_1$ 的输入：

$$\Gamma_1 = E(L) \oplus K = 35e5413d4c5e$$

使用表 8.8 得到$(S\text{-box})_1$ 的 32 比特输出 Ω_1 为

$$\Omega_1 = da5d1397$$

根据表 8.9 得到 Ω_1 置换后的数据为

$$P(\Omega_1) = e6abb863$$

将 32 比特的 $P(\Omega_1)$扩展到 48 比特，得到

$$E[P(\Omega_1)] = f0d557df0307$$

另一方面，$E(R)$与 48 比特 K 异或，所得结果作为$(S\text{-box})_2$ 的输入：

$$\Gamma_2 = E(R) \oplus K$$
$$= \text{e098b05d670d}$$

使用表 8.8 得到(S-box)$_2$ 的 32 比特输出 Ω_2 为

$$\Omega_2 = \text{3f6fa467}$$

根据表 8.9 得到 Ω_2 置换后的数据为

$$P(\Omega_2) = \text{c15b5efe}$$

若使用表 8.7 将 32 比特的 $P(\Omega_2)$扩展到 48 比特，可以得到

$$E[P(\Omega_2)] = \text{602af6afd7fd}$$

最后，串联 $E[P(\Omega_1)]$和 $E[P(\Omega_2)]$得到

$$E[P(\Omega_1)] \| E[P(\Omega_2)] = \text{f0d557df0307602af6afd7fd}$$

先将这个十六进制的数转换成二进制数，然后在这个二进制序列两端分别去掉 12 个比特，剩余比特序列如下：

0101 0101 0111 1101 1111 0000 0011 0000 0111

0110 0000 0010 1010 1111 0110 1010 1111 1101

将抽取规则应用于这个序列，每 4 个比特取 1 个比特，得到 18 比特鉴权数据，显示如下：

AUTHR = 111110101000010011

例 8.10　参考图8.18，扩展 $E[P(\Omega_1)]$和 $E[P(\Omega_1)]$这个步骤可以省略，$P(\Omega_1)$直接和 $P(\Omega_2)$串联，即 $P(\Omega_1) \| P(\Omega_2)$。

情况 1. 对于 176 比特的已填充消息：

$$P(\Omega_1) \| P(\Omega_2) = \text{af9c4880a72c1b3a}$$

先将这个十六进制的数转换成二进制数，然后在这个二进制序列两端分别去掉 5 个比特，剩余比特序列如下：

* 111 1001 1100 0100 1000 1000 0000

1010 0111 0010 1100 0001 1011 001 *

将抽取规则应用于这个序列，每 3 个比特取 1 个比特，得到 18 比特鉴权数据，显示如下：

AUTHR = 101000000110000111

情况 2. 对于 192 比特的填充后消息：

$$P(\Omega_1) \| P(\Omega_2) = \text{e6abb863c15b5efe}$$

先将这个十六进制的数转换成二进制数，然后在这个二进制序列两端分别去掉 5 个比特，剩余比特序列如下：

* 110 1010 1011 1011 1000 0110 0011

1100 0001 0101 1011 0101 1110 111 *

将抽取规则应用于这个序列，每 3 个比特取 1 个比特，得到 18 比特鉴权数据显示如下：

AUTHR = 010110101000111101

8.11.6　修正的 CBC 方案

至此，可以看出有很多方法可以用于鉴权数据的计算，修正的 CBC 模式也是其中之一，图 8.19 是这个方案的块状图，这个修正的 CBC 方案在概念思想上与在 8.11.4 节中讨论过的 DM 方案有

相似之处。

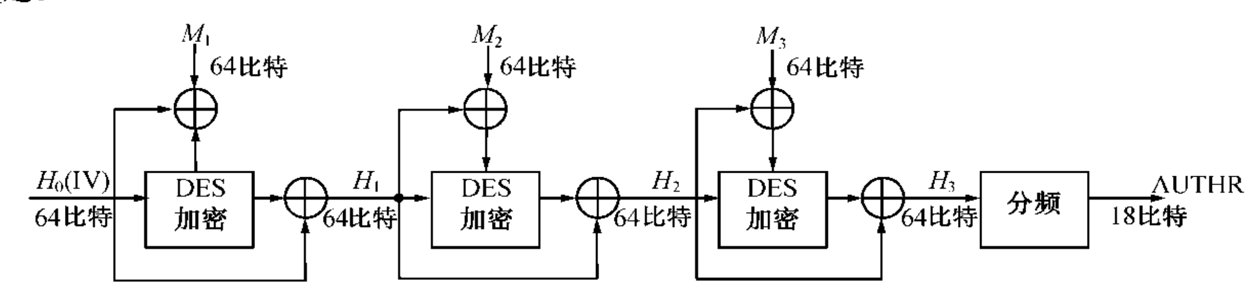


图 8.19 用改进的 CBC 方案计算 AUTHR

将要进行 Hash 转换的 192 比特的已填充消息分成三个消息块，即 M_1，M_2 和 M_3，其中 M_i，$1 \leqslant i \leqslant 3$，均为 64 比特长。

设 H_0 为初始向量（IV），加密步骤如下所示：

$$H_1 = E_{H_0 \oplus M_1}(H_0) \oplus H_0$$

$$H_2 = E_{H_1 \oplus M_2}(H_1) \oplus H_1$$

$$H_3 = E_{H_2 \oplus M_3}(H_2) \oplus H_2$$

这里 $H_i \oplus M_j$ 表示加密密钥。

既然 H_3 包含 64 个比特，从它的左边和右边各去掉 5 个比特后得到 54 比特的 H_3 简化序列。

从简化后的序列中每 3 个比特抽取 1 比特，就可以得到期望的 18 比特鉴权数据。

18 比特 AUTHR 的计算与 DM 方案中的计算完全相似，因此留给读者自己练习。

8.12 SSD 生成

SSD 是存储于移动台半永久存储器中的 128 比特共享加密数据，如图 8.4 所示，SSD 被分成两个不同的子集：SSD-A 和 SSD-B。SSD-A 用于鉴权，SSD-B 用于话音保密和信息加密。

如图 8.20 所示，SSD 的更新是通过 SSD 生成程序进行的，A 钥是分配给移动台的 64 比特长的数据，并且它仅对移动台和相关的归属位置寄存器/鉴权中心（HLR/AC）是可知的。SSD 更新只在移动台和相关的 HLR/AC 中执行，而不是在正在服务的基站中执行。

SSD 更新过程执行如下：

- 基站通过寻呼信道或前向业务信道的 SSD 更新消息向移动台发送 SSD 更新指令，其中 RANDSSD 字段中包含的 56 比特随机数与 HLR/AC 中计算的相同。
- 收到 SSD 更新消息后，移动台设置 SSD 生成程序的输入参数，执行 SSD 生成程序，计算出 SSD-A-NEW 等于输出结果的高位 64 比特，SSD-B-NEW 等于低位 64 比特。

例 8.11　当收到 SSD 更新消息后，移动台设置输入参数（见图8.20 所示）：

RANDSSD（56 比特），ESN（32 比特），A 钥（64 比特）及填充比特（40 比特）

通过添加 40 个填充比特将 152 比特长的消息扩展到 192 比特。

参考图 8.14，假设

M_1（64 比特输入密钥）：7a138b2524af17c3

从而得到 K（48 比特 DES 密钥）：058c4517a7a2

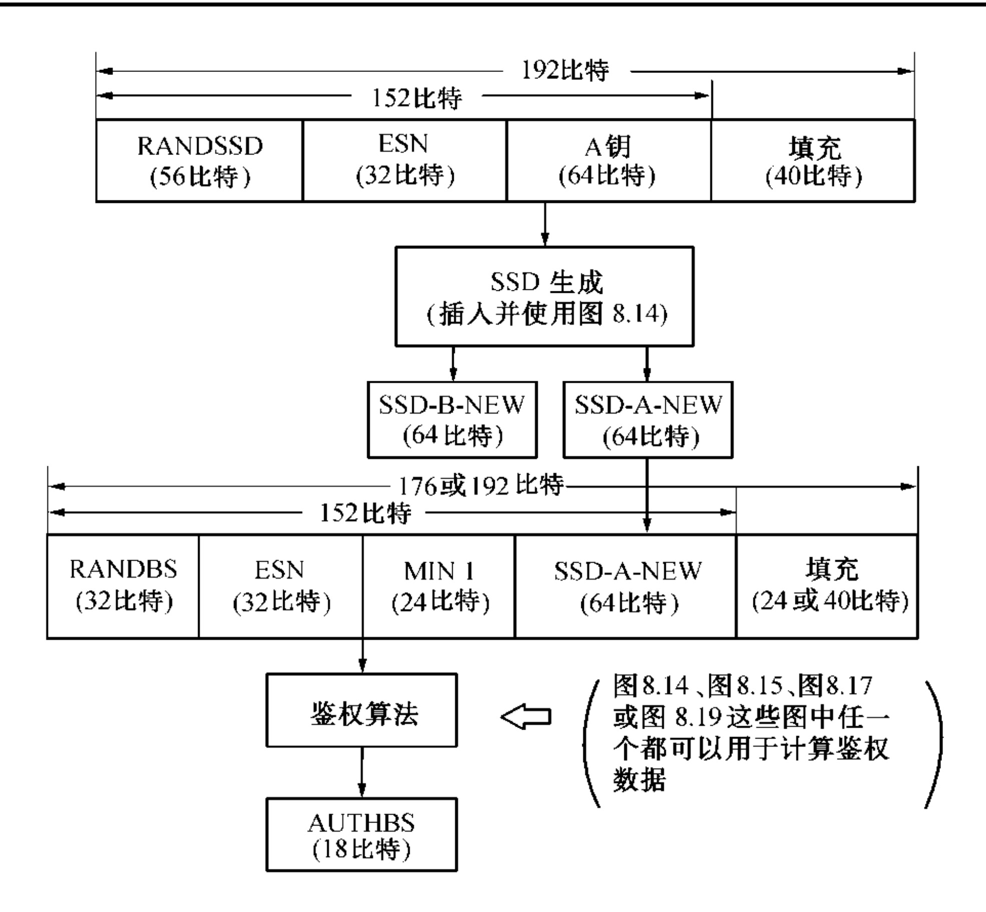


图 8.20　SSD 的生成和 AUTHBS 的计算

$$M_2\text{(64 比特)}: 17b439a12f51c5a8$$

$$M_3\text{(64 比特)}: 51cb360000000000$$

分别将(S-box)$_i$ 的输出进行置换后得到 $P(\Omega_i)$，$i = 1,2,3,4$ 的值如下：

$$P(\Omega_1) = a34d397f$$

$$P(\Omega_2) = 91193e8e$$

$$P(\Omega_3) = c3cc8226$$

$$P(\Omega_4) = 02234098$$

S-box 最后的输出数据是这四个数据块的集合，将它们串联就得到 128 比特的 $P(\Omega)$：

$$P(\Omega) = P(\Omega_1) \parallel P(\Omega_2) \parallel P(\Omega_3) \parallel P(\Omega_4)$$
$$= (a34d397f91193e8ec3cc822602234098)$$

从这我们可以得到

SSD-A-NEW ＝ a34d397f91193e8e(注：此处原文为 SSD-B-NEW，可能有误)

SSD-B-NEW ＝ c3cc822602234098(注：此处原文为 SSD-A-NEW，可能有误)

可以通过已在 8.11 节中提出的各种各样的鉴权算法计算 18 比特的消息 AUTHBS，在图 8.20 中，AUTHBS 的计算应当结合 SSD-A-NEW 进行。

8.12.1　使用 MD5 算法计算 SSD-A 和 SSD-B

MD5 是 MD4 的修正版，它们在设计思想上是相似的，MD5 接受一个任意长度的消息，产生 128 比特的消息摘要作为输出，它在设计上保守一些，但是速度比 MD4 稍慢一点，MD5 处理以 512 比特块形式输入的消息，这样可以将它分成 16 个 32 比特的子块，消息摘要是由 4 个 32 比特块串联而

成的 128 比特的 Hash 值。既然 SSD 更新是通过存储在移动台的 128 比特的模式实现的，那它就可以分成两个不同的子集，即 SSD-A-NEW 和 SSD-B-NEW。

计算输入消息的消息摘要的步骤如下：

1. 添加填充比特

填充消息使得它的比特长度模 512 为 448，也就是已填充的消息正好比 512 的倍数少 64 个比特。

在消息的填充过程中，先是添加一个 1，然后添加足够多的 0，使得总长度模 512 为 448（$=512-64$）。

2. 初始化 MD 寄存器

四个 32 比特寄存器（A,B,C,D）用于消息摘要的计算，给这些寄存器赋初值（用十六进制表示）

A:01 23 45 67

B:89 ab cd ef

C:fe dc ba 98

D:76 54 32 10

将这四个变量的值复制到四个不同的变量中：$AA=A$，$BB=B$，$CC=C$，$DD=D$。

3. 定义四个辅助函数（F,G,H,I）

F,G,H,I 是基本的 MD5 函数。

它们是四个非线性函数，一轮运算用一个。

$$F(X,Y,Z)=X\cdot Y+X'\cdot Z$$

$$G(X,Y,Z)=X\cdot Z+Y\cdot Z'$$

$$H(X,Y,Z)=Y\oplus Y\oplus Z$$

$$I(X,Y,Z)=Y\oplus(X+Z')$$

在这里，说明一下这些符号的表示：$X\cdot Y$ 表示 X 与 Y 按位逻辑"与"，$X+Y$ 表示 X 与 Y 按位逻辑"或"，X' 表示 X 的按位逻辑"非"，即 NOT(X)，$X\oplus Y$ 表示 X 与 Y 按位逻辑"异或"。

以这样的方式设计这四个非线性函数的目的在于：如果 X,Y,Z 的比特是独立无偏的，那么在每个比特位上，函数 F 相当于一个条件句：if X then Y else Z。函数 G,H,I 与函数 F 相似，因为它们相当于 X,Y 和 Z 的位逻辑并联。注意，H 是输入的按位逻辑异或函数。

四个非线性函数（F,G,H,I）计算的真值表见表 8.14。

表 8.14 四个非线性函数的真值表

X Y Z	F G H I
000	0000
001	1010
010	0110
011	1001
100	0011
101	0101
110	1100
111	1110

4. 用于第 1,2,3,4 轮中的 FF,GG,HH 和 II 变换

   若 $M[k]$,$0\leqslant k\leqslant 15$,表示消息的第 k 个子块,$<<<s$ 表示左移 s 比特,则定义如下四个运算:

   $FF(a,b,c,d,M[k],s,i):a = b+((a+F(b,c,d)+M[k]+T[i])<<<s)$

   $GG(a,b,c,d,M[k],s,i):a = b+((a+G(b,c,d)+M[k]+T[i])<<<s)$

   $HH(a,b,c,d,M[k],s,i):a = b+((a+H(b,c,d)+M[k]+T[i])<<<s)$

   $II(a,b,c,d,M[k],s,i):a = b+((a+I(b,c,d)+M[k]+T[i])<<<s)$

   计算中用到一个含 64 个元素的表 $T[i]$,$i = 1,2,\cdots,64$,这个表是根据正弦函数构造的,$T[i]$表示表中的第 i 个元素,它等于 $\sin(i)$绝对值的 4 294 967 296 部的整数部分,其中 i 是弧度数:

   $$T[i] = \text{integer part of }[2^{32} * |\sin(i)|]$$

   这里,$0\leqslant|\sin(i)|\leqslant 1$ 且 $0\leqslant 2^{32} * |\sin(i)|\leqslant 2^{32}$。

   T$[i]$,$1\leqslant i\leqslant 64$,的计算结果已经制成表格,见表 8.15。

5. 四轮的计算过程(64 步)

   每一轮包含 16 个运算,每个运算都是在 A,B,C,D 中的三个上执行一个非线性函数。

   下面是第 1,2,3,4 轮中的 FF,GG,HH 和 II 变换

   第 1 轮

   设 $FF(a,b,c,d,M[k],s,i)$表示运算

   $$a = b+((a+F(b,c,d)+M[k]+T[i])<<<s)$$

   然后计算下面 16 个运算:

   $FF(a,b,c,d,M[0],7,1)$,$FF(d,a,b,c,M[1],12,2)$,$FF(c,d,a,b,M[2],17,3)$
   $FF(b,c,d,a,M[3],22,4)$,$FF(a,b,c,d,M[4],7,5)$,$FF(d,a,b,c,M[5],12,6)$,
   $FF(c,d,a,b,M[6],17,7)$,$FF(b,c,d,a,M[7],22,8)$,$FF(a,b,c,d,M[8],7,9)$,
   $FF(d,a,b,c,M[9],12,10)$,$FF(c,d,a,b,M[10],17,11)$,$FF(b,c,d,a,M[11],22,12)$,
   $FF(a,b,c,d,M[12],7,13)$,$FF(d,a,b,c,M[13],12,14)$,$FF(c,d,a,b,M[14],17,15)$,
   $FF(b,c,d,a,M[15],22,16)$

表 8.15　$T[i]$的计算,$1\leqslant i\leqslant 64$

$T[1]$ = d76aa478	$T[17]$ = f61e2562	$T[33]$ = fffa3942	$T[49]$ = f4292244
$T[2]$ = e8c7b756	$T[18]$ = c040b340	$T[34]$ = 8771f681	$T[50]$ = 432aff97
$T[3]$ = 242070db	$T[19]$ = 265e5a51	$T[35]$ = 69d96122	$T[51]$ = ab9423a7
$T[4]$ = c1bdceee	$T[20]$ = e9b6c7aa	$T[36]$ = fde5380c	$T[52]$ = fc93a039
$T[5]$ = f57c0faf	$T[21]$ = d62f105d	$T[37]$ = a4beea44	$T[53]$ = 655b59c3
$T[6]$ = 4787c62a	$T[22]$ = 02441453	$T[38]$ = 4bdecfa9	$T[54]$ = 8f0ccc92
$T[7]$ = a8304613	$T[23]$ = d8a1e681	$T[39]$ = f6bb4b60	$T[55]$ = ffeff47d
$T[8]$ = fd469501	$T[24]$ = e7d3fbc8	$T[40]$ = bebfbc70	$T[56]$ = 85845dd1
$T[9]$ = 698098d8	$T[25]$ = 21e1cde6	$T[41]$ = 289b7ec6	$T[57]$ = 6fa87e4f
$T[10]$ = 8b44f7af	$T[26]$ = c33707d6	$T[42]$ = eaa127fa	$T[58]$ = fe2ce6e0
$T[11]$ = ffff5bb1	$T[27]$ = f4d50d87	$T[43]$ = d4ef3085	$T[59]$ = a3014314
$T[12]$ = 895cd7be	$T[28]$ = 455a14ed	$T[44]$ = 04881d05	$T[60]$ = 4e0811a1
$T[13]$ = 6b901122	$T[29]$ = a9e3e905	$T[45]$ = d9d4d039	$T[61]$ = f7537e82
$T[14]$ = fd987193	$T[30]$ = fcefa3f8	$T[46]$ = e6db99e5	$T[62]$ = bd3af235
$T[15]$ = a679438e	$T[31]$ = 676f02d9	$T[47]$ = 1fa27cf8	$T[63]$ = 2ad7d2bb
$T[16]$ = 49b40821	$T[32]$ = 8d2a4c8a	$T[48]$ = c4ac5665	$T[64]$ = eb86d391

第 2 轮

设 $GG(a,b,c,d,M[k],s,i)$表示运算

$$a = b + ((a + G(b,c,d) + M[k] + T[i]) <<< s)$$

然后计算下面 16 个运算：

$GG(a,b,c,d,M[1],5,17)$，$GG(d,a,b,c,M[6],9,18)$，$GG(c,d,a,b,M[11],14,19)$，
$GG(b,c,d,a,M[0],20,20)$，$GG(a,b,c,d,M[5],5,21)$，$GG(d,a,b,c,M[10],9,22)$，
$GG(c,d,a,b,M[15],14,23)$，$GG(b,c,d,a,M[4],20,24)$，$GG(a,b,c,d,M[9],5,25)$，
$GG(d,a,b,c,M[14],9,26)$，$GG(c,d,a,b,M[3],14,27)$，$GG(b,c,d,a,M[8],20,28)$，
$GG(a,b,c,d,M[13],5,29)$，$GG(d,a,b,c,M[2],9,30)$，$GG(c,d,a,b,M[7],14,31)$，
$GG(b,c,d,a,M[12],20,32)$

第 3 轮

设 $HH(a,b,c,d,M[k],s,i)$表示运算

$$a = b + ((a + H(b,c,d) + M[k] + T[i]) <<< s)$$

然后计算下面 16 个运算：

$HH(a,b,c,d,M[5],4,33)$，$HH(d,a,b,c,M[8],11,34)$，$HH(c,d,a,b,M[11],16,35)$，
$HH(b,c,d,a,M[14],23,36)$，$HH(a,b,c,d,M[1],4,37)$，$HH(d,a,b,c,M[4],11,38)$，
$HH(c,d,a,b,M[7],16,39)$，$HH(b,c,d,a,M[10],23,40)$，$HH(a,b,c,d,M[13],4,41)$，
$HH(d,a,b,c,M[0],11,42)$，$HH(c,d,a,b,M[3],16,43)$，$HH(b,c,d,a,M[6],23,44)$，
$HH(a,b,c,d,M[9],4,45)$，$HH(d,a,b,c,M[2],11,46)$，$HH(c,d,a,b,M[15],16,47)$，
$HH(b,c,d,a,M[2],23,48)$

第 4 轮

设 $II(a,b,c,d,M[k],s,i)$表示运算

$$a = b + ((a + I(b,c,d) + M[k] + T[i]) <<< s)$$

然后计算下面 16 个运算：

$II(a,b,c,d,M[0],6,49)$，$II(d,a,b,c,M[7],10,50)$，$II(c,d,a,b,M[14],15,51)$，
$II(b,c,d,a,M[5],21,52)$，$II(a,b,c,d,M[12],6,53)$，$II(d,a,b,c,M[3],10,54)$，
$II(c,d,a,b,M[10],15,53)$，$II(b,c,d,a,M[1],21,56)$，$II(a,b,c,d,M[8],6,57)$，
$II(d,a,b,c,M[15],10,58)$，$II(c,d,a,b,M[6],15,59)$，$II(b,c,d,a,M[13],21,60)$，
$II(a,b,c,d,M[4],6,61)$，$II(d,a,b,c,M[11],10,62)$，$II(c,d,a,b,M[2],15,63)$，
$II(b,c,d,a,M[9],21,64)$

完成上述所有步骤后，将 A,B,C,D 加到它们各自的增量 AA,BB,CC,DD 上，如下所示：

$$A = A + AA, B = B + BB$$

$$C = C + CC, D = D + DD$$

然后继续处理剩下的数据块，最后的结果是 A,B,C,D 的串联。

例 8.12　设初始寄存器的值为：

$$A = 67452301,\ B = \text{efcdab89}$$

$$C = \text{98badcfe},\ D = 10325476$$

已填充的 512 比特的消息由 152 比特的原始消息加上 360 比特的填充构成，如下所示：

已填充的消息（512 比特） = 原始消息（152 比特）+ 填充（360 比特）

即

7a138b25 24af17c3 17b439a1 2f51c5a8

51cb3600 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

1. 第 1 轮计算（$M[0]$，$T[1]$，$s = 7$）

$$a = b + ((a + F(b,c,d) + M[0] + T[1]) <<< s)$$

其中，U<<<s 表示将 U 循环左移 s 个位置。

使用表 8.14（注：此处原文为 8.12，可能有误），$F(b,c,d)$的计算如下所示：

b：1110 1111 1100 1101 1010 1011 1000 1001

c：1001 1000 1011 1010 1101 1100 1111 1110

d：0001 0000 0011 0010 0101 0100 0111 0110

$F(b,c,d)$：1001 1000 1011 1010 1101 1100 1111 1110

9　8　b　a　d　c　f　e

计算 U $= (a + F(b,c,d) + M[0] + T[1]) <<< s$，$s = 7$

a：67452301

$F(b,c,d)$：98badcfe

$M[0]$：7a138b25

$T[1]$：d76aa478

U：517e2f9c

U = 517e2f9c<<<7

= 0101 0001 0111 1110 0010 1111 1001 1100

U<<<7 表示将 U 循环左移 7 位后得到的 32 比特值，所以移位后的值为

U′：1011 1111 0001 0111 1100 1110 0010 1000

b　f　1　7　c　e　2　8

由 $a = b + \mathrm{U}'$，得到

b：efcdab89

U′：bf17ce28

---

a：aee57961

因此 $FF(a,b,c,d,M[0],7,1)$计算结果如下

aee57961，efcdab89，98badcfe，10325476

第 1 轮所有的 FF 变换都可以类似地计算出来，16 次运算的结果如下：

[1]　aee579b1, efcdab89, 98badcfe, 10325476

[2]　aee579b1, efcdab89, 98badcfe, 3bfb6779

[3]　aee579b1, efcdab89, 1e52ee63, 3bfb6779

[4]　aee579b1,2279e391, 1e52ee63, 3bfb6779

[5]　224cc819,2279e391, 1e52ee63, 3bfb6779

[6]　224cc819,2279e391, 1e52ee63, 38e34369

[7]　224cc819,2279e391, 249e0639, 38e34369

[8]　224cc819, 5526aa0a, 249e0639, 38e34369

2. 第 2 轮计算（$M[1]$，$T[17]$，$s=5$）

$$a = b + ((a + G(b,c,d) + M[1] + T[17]) <<< s)$$

设 $V=(a+G(b,c,d)+M[1]+T[17]) <<< s, s=5$

这里，a = 21bfac5f，b = 25a998d6，c = 76c0395e，d = 7f92ed5d

$M[1]$ = 24af17c3，$T[17]$ = f61e2562

使用表 8.14（注：此处原文为 8.12，可能有误），$G(b,c,d)$的计算如下所示：

```
          b:0010 0101 1010 1001 1001 1000 1101 0110
          c:0111 0110 1100 0000 0011 1001 0101 1110
          d:0111 1111 1001 0010 1110 1101 0101 1101
---------------------------------------------------
 G(b,c,d):0010 0101 1100 0000 1001 1000 0101 0110
            2    5    c    0    9    8    5    6
```

计算 $V = a + G(b,c,d) + M[1] + T[17] <<< 5$

```
       a:21bfac5f
G(b,c,d):25c09856
    M[1]:24af17c3
   T[17]:f61e2562
-----------------
       V:624d81da
```

因此，V = 624d81da <<< 5

= 0110 0010 0100 1101 1000 0001 1101 1010

将 V 循环左移 5 位后得到

```
V' = 0100 1001 1011 0000 0011 1011 0100 1100
       4    9    b    0    3    b    4    c
```

由 $a = b + V'$，得

```
 b:25a998d6
V':49b03b4c
-----------
 a:6f59d422
```

这样通过下面的 16 次运算，第 2 轮所有的 GG 变换都可以类似地计算出来，结果如下：

[1] 6f59d422, 25a998d6, 76c0395e, 7f92ed5d
[2] 6f59d422, 25a998d6, 76c0395e, eaed1370
[3] 6f59d422, 25a998d6, e4f4d575, eaed1370
[4] 6f59d422, 164bbc01, e4f4d575, eaed1370
[5] 92a2cc8a, 164bbc01, e4f4d575, eaed1370
[6] 92a2cc8a, 164bbc01, e4f4d575, 4cca5389
[7] 92a2cc8a, 164bbc01, 8feaa429, 4cca5389
[8] 92a2cc8a, dd0c9181, 8feaa429, 4cca5389
[9] 52b693b1, dd0c9181, 8feaa429, 4cca5389
[10] 52b693b1, dd0c9181, 8feaa429, a2909476
[11] 52b693b1, dd0c9181, a4f2a227, a2909476
[12] 52b693b1, 724ed3b9, a4f2a227, a2909476
[13] d1f4b09d, 724ed3b9, a4f2a227, a2909476
[14] d1f4b09d, 724ed3b9, a4f2a227, 34bc09b1
[15] d1f4b09d, 724ed3b9, a4f2a227, 34bc09b1
[16] d1f4b09d, 3e23b74d, a4f2a227, 34bc09b1

3. 第 3 轮计算($M[5]$,$T[33]$,$s=4$)

$$a = b+((a+H(b,c,d)+M[5]+T[33])<<<s)$$

这里,a = d1f4b09d,b = 3e23b74d,c = a6697938,d = 34bc09b1

$M[5]$ = 00000000,$T[33]$ = fffa3942,s = 4

设 $W = (a+H(b,c,d)+M[5]+T[33])<<<4$

使用表 8.14(注:此处原文为 8.12,可能有误),$H(b,c,d)$的计算如下所示:

b:0011 1110 0010 0011 1011 0111 0100 1101
c:1010 0110 0110 1001 0111 1001 0011 1000
d:0011 0100 1011 1100 0000 1001 1011 0001

$H(b,c,d)$:1010 1100 1111 0110 1100 0111 1100 0100
a　c　f　6　c　7　c　4

计算 $W = a+H(b,c,d)+M[5]+T[33]<<<4$

a:d1f4b09d
$H(b,c,d)$:acf6c7c4(注:此处原文为 $G(b,c,d)$,可能有误)
$M[5]$:00000000(注:此处原文为 $M[1]$,可能有误)
$T[33]$:fffa3942(注:此处原文为 $T[17]$,可能有误)

W:7ee5b1a3

因此,W = 7ee5b1a3 <<<4
= 0111 1110 1110 0101 1011 0001 1010 0011

将 W 循环左移 4 位后得到

W' = 1110 1110 0101 1011 0001 1010 0011 0111
e　e　5　b　1　a　3　7

由 $a = b+W'$,得

b:3e23b74d
W':ee5b1a37

---

a:2c7ed184

因此 $HH(a,b,c,d,M[5],4,33)$计算结果如下

2c7ed184,3e23b74d,a6697938,34bc09b1

这样通过下面的 16 次运算,第 3 轮所有的 HH 变换都可以类似地计算出来,结果如下:

[1]　2c7ed184, 3e23b74d, a6697938, 34bc09b1
[2]　2c7ed184, 3e23b74d, a6697938, 3d7fed07
[3]　2c7ed184, 3e23b74d, d3a8c930, 3d7fed07
[4]　2c7ed184, 5a2822a2, d3a8c930, 3d7fed07
[5]　8e5c4ac4, 5a2822a2, d3a8c930, 3d7fed07
[6]　8e5c4ac4, 5a2822a2, d3a8c930, 81f53790
[7]　8e5c4ac4, 5a2822a2, 6823d52c, 81f53790
[8]　8e5c4ac4, f9a0e2ae, 6823d52c, 81f53790
[9]　19151af2, f9a0e2ae, 6823d52c, 81f53790
[10]　19151af2, f9a0e2ae, 6823d52c, 19d6166c
[11]　19151af2, f9a0e2ae, d35f8234, 19d6166c
[12]　19151af2, 42486547, d35f8234, 19d6166c
[13]　fd0fb1ee, 42486547, d35f8234, 19d6166c
[14]　fd0fb1ee, 42486547, d35f8234, 4d472554
[15]　fd0fb1ee, 42486547, 3e710a56, 4d472554

4. 第 4 轮计算($M[0], T[49], s = 6$)

$$a = b + ((a + I(b,c,d) + M[0] + T[49]) <<< s), s=6$$

这里，a = fd0fb1ee，b = 5b477ba0，c = 3e710a56，d = 4d472554

$M[0]$ = 7a138b25，$T[49]$ = f4292244

设 $Z = (a + I(b,c,d) + M[0] + T[49]) <<< 6$

使用表 8.14(注：此处原文为 8.12，可能有误)，$I(b,c,d)$的计算如下所示：

b:0101 1011 0100 0111 0111 1011 1010 0000
c:0011 1110 0111 0001 0000 1010 0101 0110
d:0100 1101 0100 0111 0010 0101 0101 0100

$I(b,c,d)$:1100 0101 1000 1110 1111 0001 1111 1101
c 5 8 e f 1 f d

计算 $Z = a + I(b,c,d) + M[0] + T[49] <<< 6$

a:fd0fb1ee
$I(b,c,d)$:c58ef1fd
$M[0]$:7a138b25
$T[49]$:f4292244

Z:30db5154

因此，Z = 30db5154 $<<<6$
= 0011 0000 1101 1011 0101 0001 0101 0100

将 Z 循环左移 6 位后得到

Z' = 0011 0110 1101 0100 0101 0101 0000 1100
3 6 d 4 5 5 0 c

由 $a = b + Z'$，得

b:5b477ba0
Z':36d4550c

---

a:921bd0ac

因此 $II(a,b,c,d,M[0],6,33)$计算结果如下

921bd0ac，5b477ba0，3e710a56，4d472554

这样通过下面的 16 次运算，第 4 轮所有的 II 变换都可以类似地计算出来，结果如下：

[1] 921bd0ac, 5b477ba0, 3e710a56, 4d472554
[2] 921bd0ac, 5b477ba0, 3e710a56, bce7b111
[3] 921bd0ac, 5b477ba0, 86dfbd85, bce7b111
[4] 921bd0ac, dad51bea, 86dfbd85, bce7b111
[5] f958527f, dad51bea, 86dfbd85, bce7b111
[6] f958527f, dad51bea, 86dfbd85, c78bd4fa
[7] f958527f, dad51bea, 73cd27db, c78bd4fa
[8] f958527f, 07b3d57c, 73cd27db, c78bd4fa
[9] 569e32a9, 07b3d57c, 73cd27db, c78bd4fa
[10] 569e32a9, 07b3d57c, 73cd27db, 6e4ce124
[11] 569e32a9, 07b3d57c, 5c8bc0f4, 6e4ce124
[12] 569e32a9, da88ed56, 5c8bc0f4, 6e4ce124

四个 32 比特的缓冲区(A,B,C,D)用于计算消息摘要，每个都是 32 比特的寄存器，给这些寄存器赋下列初值：

$$aa = 67452301,\ bb = \text{efcdab89}$$
$$cc = \text{98badcfe},\ dd = 10325476$$

H 变换的最后一次运算[16]的结果是

$$a = \text{95d282df},\ b = \text{cd5df50b}$$
$$c = \text{63ac7a25},\ d = \text{ae6a0d43}$$

SSD 更新后的数据，即 SSD-A-NEW 和 SSD-B-NEW 的产生如下：

a:95d282df	b:cd5df50b
aa:67452301	bb:efcdab89
A:fd17a5e0	B:bd2ba094

SSD-A-NEW = fd17a5e0bd2ba094

同样，

c:63ac7a25	d:ae6a0d43
cc:98badcfe	dd:10325476
C:fc675723	D:be9c61b9

SSD-B-NEW = fc675723be9c61b9

8.13　消息加密和安全

为了保护用户的敏感信息，必须对所选业务信道中信令消息的某些字段进行加密，但是，如果不执行鉴权过程，将不对消息进行加密。每个反向业务信道的消息中包含一个字段，用来标志消息创建时激活的加密模式。

8.13.1　非线性组合生成的加密密钥

m 级 LFSR 由抽头 g_1 到 g_m 和 m 个存储器单元构成，由它产生的密钥比特序列用做加密密钥是不合适的，因为通过判断抽头和移位寄存器初始值的方法，只要得到明文或密文的 $2m$ 个比特就足够破译出密钥比特流。

当采用周期为 $p = 2^m - 1$ 的 m 级 LFSR 产生一个 PN 序列时，称它为最大长度移位寄存器。

一些密码学家提议将由 LFSR 产生的 PN 序列结合一些非线性函数后应用于密码学，很多人建议采用由一个非线性函数结合若干个最大长度移位寄存器构成的密钥流发生器。将非线性组合技术用于 LFSR 序列后显示出某些统计上的特性，使它能够经受任何密码攻击。

下面讨论 m 级 LFSR，其生成的二进制序列适用于非线性滤波函数 f，如图 8.21 所示。

对密码分析而言，非线性密钥流应当是安全的，为了得到生成密钥序列的高不可预测性，我们将发展二进制序列的一个非线性理论，这个理论直接反映了布尔函数中 k 个变量的乘积和。设 $f(x)$ 为变量 x_i 的任一非线性函数，$1 \leqslant i \leqslant n$，则

$$f(x) = a_1x_1 + a_2x_2 + \cdots + a_nx_n + a_{12}x_1x_2 + \cdots + a_{n-1,n}x_{n-1}x_n$$

$$+a_{123}x_1x_2x_3+\cdots+a_{n-2,n-1,n}x_{n-2}x_{n-1}x_n$$
$$\cdots\cdots\cdots\cdots$$
$$+a_{12\cdots n}x_1x_2\cdots x_n$$

这里，变量的一个乘积叫做第 k 阶乘积，因此 $f(x)$ 的阶数是它的乘积项中阶数最大的。

例 8.13 图8.21给出了一个可以应用于 CDMA 移动通信系统的 LFSR，它是结合了非线性滤波函数 $f(x)$ 的 32 级 LFSR。假设，最大长度 LFSR 的生成多项式是 $g(x)=1+x+x^2+x^{22}+x^{32}$。

存储于移动台半永久存储器中的 128 比特 SSD 被分成两个不同的子集：SSD-A 和 SSD-B。64 比特的 SSD-A 用于鉴权，64 比特的 SSD-B 用于 CDMA 消息保密。

64 比特的 SSD-B 分成两半（即各 32 比特），高位 32 比特分配给 32 级 LFSR 作为初始值；低位 32 比特分配给非线性函数（见图 8.21 所示），低位的 32 比特再平均分成四个 8 比特组。

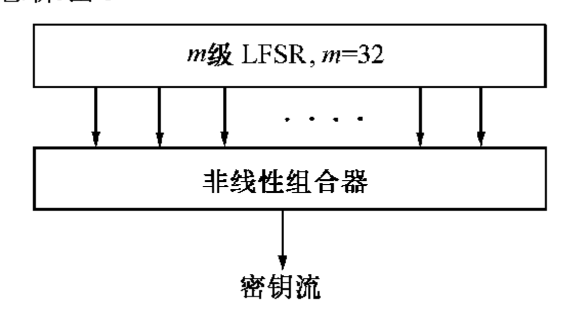


图 8.21 用非线性组合器过滤的密钥比特生成器

假设64 比特的 SSD-B 为 30a7f415 83c7519a（十六进制），则高位 32 比特初始值为 30a7f415（十六进制）＝00110000101001111111010000010101（二进制）；而用于非线性函数的低位 32 比特为 83c7519a（十六进制）＝10000011110001110101000110011010（二进制）。

为将低位的 32 比特用做非线性函数 $f(x)$ 的系数，把它分成等长的四个 8 比特组，如下所示：

第一个非线性组合器的系数：$(a_1, a_2, a_3, ...)$

83（十六进制）＝10000011（二进制）$\rightarrow x_1+x_7+x_8$

第二个非线性组合器的系数：$(a_{12}, a_{13}, a_{14}, ...)$

c7（十六进制）＝11000111（二进制）$\rightarrow x_1x_2+x_1x_3+x_1x_7+x_1x_8+x_1x_9$

第三个非线性组合器的系数：$(a_{123}, a_{124}, a_{125}, a_{126}, ...)$

51（十六进制）＝01010001（二进制）$\rightarrow x_1x_2x_4+x_1x_2x_6+x_1x_2x_{10}$

第四个非线性组合器的系数：$(a_{1234}, a_{1235}, a_{1236}, ...)$

9a（十六进制）＝10011010（二进制）$\rightarrow x_1x_2x_3x_4+x_1x_2x_3x_7+x_1x_2x_3x_8+x_1x_2x_3x_{10}$

这样得到非线性函数 $f(x)$ 为

$$f(x)=x_1+x_7+x_8+x_1x_2+x_1x_3+x_1x_7+x_1x_8+x_1x_9+x_1x_2x_4+x_1x_2x_6+x_1x_2x_{10}+x_1x_2x_3x_4$$
$$+x_1x_2x_3x_7+x_1x_2x_3x_8+x_1x_2x_3x_{10}$$

结合高位 32 比特的初始值，相应的 LFSR 的值为

$$x_1=0, x_2=0, x_3=1, x_4=1, x_5=0,$$
$$x_6=0, x_7=0, x_8=0, x_9=1, x_{10}=0。$$

利用 LFSR 的值，确定非线性组合器的系数：

一次项系数：$x_1+x_7+x_8=0\oplus 0\oplus 0=0$

二次项系数：$x_1x_2+x_1x_3+x_1x_7+x_1x_8+x_1x_9=0\oplus 0\oplus 0\oplus 0\oplus 0=0$

三次项系数：$x_1x_2x_4+x_1x_2x_6+x_1x_2x_{10}=0\oplus 0\oplus 0=0$

四次项系数：$x_1x_2x_3x_4+x_1x_2x_3x_7+x_1x_2x_3x_8+x_1x_2x_3x_{10}=0\oplus 0\oplus 0\oplus 0=0$

因此，非线性组合器的输出数据为 $0\oplus 0\oplus 0\oplus 0=0$。

将 32 比特的初始值用到图 8.21 中，并且 LFSR 右移位，则计算出从非线性组合器中输出的 PN 序列应如表 8.16 所示。

表 8.16　计算非线性组合器的输出

移位 No	LFSR 内容	组合器的输出（比特）
0	00110000101001111111101000001010l	0
1	11111000010100111111100000001010	1
2	01111100001010011111110000000101	0
3	11011111000010100111111000000010	0
4	01101111100001010011111100000001	0
5	11010111110000101001111010000000	1
…	…………	…
14	11100000011010111100000010011110	1
15	01110000000110101111000000100111 1	0
…	…………	…
28	10101011110110011000001000100110	1
29	01010101111011001100000100010011	1
30	11001010111101100110001010001001	0
…		…
49	11101110100011011110001101111101	0
50	10010111010001101111001110111110	1
…	…………	…

如下所示，可以得到表 8.16 中非线性组合器的输出 Z，它将用做密钥序列：

非线性组合器的输出序列 Z:0100010111101010110010101100 11

011100010010101001101010010011110011010 1 001110010011000000101001100101

PN 序列 Z 用做加密密钥。

例 8.14　如表8.16所示，对应于移位 No 14，LFSR 的值为

$$11100000011010111100000010011110$$

利用对应移位 No. 14 的值，LFSR 的第 i 级的值将是 $x_1=1, x_2=1, x_3=1, x_4=0, x_5=0, x_6=0, x_7=0, x_8=0, x_9=0, x_{10}=1$。因此，非线性组合器的系数计算如下：

一次项系数：$x_1+x_7+x_8=1\oplus0\oplus0=1$

二次项系数：$x_1x_2+x_1x_3+x_1x_7+x_1x_8+x_1x_9=1\oplus1\oplus0\oplus0\oplus0=0$

三次项系数：$x_1x_2x_4+x_1x_2x_6+x_1x_2x_{10}=0\oplus0\oplus1=1$

四次项系数：$x_1x_2x_3x_4+x_1x_2x_3x_7+x_1x_2x_3x_8+x_1x_2x_3x_{10}=0\oplus0\oplus0\oplus1=1$

这样，对应于移位 No. 14，组合器的输出为 $1\oplus0\oplus1\oplus1=1$。

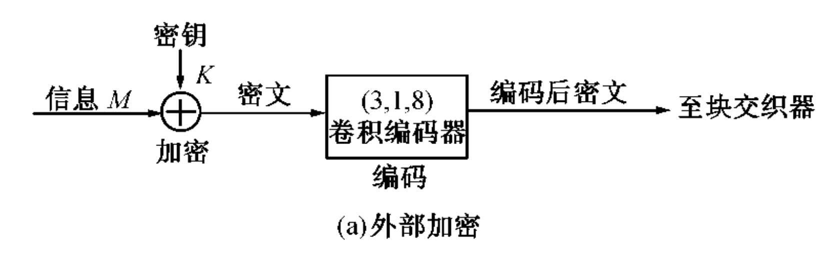
8.13.2　加密和消息安全

反向业务信道上发送的数据封装成 20 ms 的帧，在其上发送的数据帧有 9 600b/s，4 800 b/s，2 400b/s和 1 200 b/s 四个可变速率，反向业务信道用于在呼叫期间向基站发送用户和信令信息。

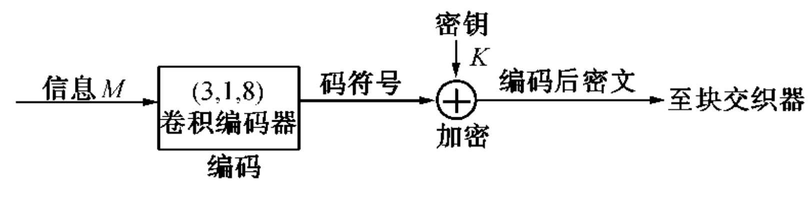
每个数据帧的持续时间是 20 ms，根据发送速率的不同，每帧可以包含信息位、CRC 校验位和

尾比特或者只包含信息位和尾比特。例如，对 4 800 b/s 的速率，一帧（96 比特）包含 80 比特的信息、8 比特的 CRC 和 8 比特的编码器尾比特。

消息加密有两种方式：外部加密和内部加密，如图 8.22 所示。



(a)外部加密



(b)内部加密

图 8.22 消息加密模式

消息信息 M 先用 8.13.1 节中图 8.21 生成的密钥流 K 进行加密，这种加密方案称为"外部加密"；第二种方案如图 8.22(b)所示，是先编码再加密，先解密后译码，这种加密方案称为"内部加密"。

例 8.15 本例考虑外部加密这种情况，见例3.12，假设 4 800 b/s 帧（20 ms）中的 80 比特信息数据为：

1010110011 1001101111 0010010100 0100110011 0011001000
1110010110 1001100110 0110011110

8.13.1 节中，从非线性密钥发生器中计算出的 PN 序列如下：

0100010111 1010101100 1010110011 0111000100 1010100110
1010010011 1100110101 0011100100 1100000010 1001100101

使用这些信息和密钥序列，得到加密后的序列为：

1110100100 0011000011 1000100111 0011110111 1001101110
0100000101 0101010011 0101111010

接着，用生成多项式 $g(x) = 1 + x + x^3 + x^4 + x^7 + x^8$ 计算在 4 800 b/s 下的 CRC，寄存器的初始状态为全 1，计算出的 CRC 如下所示：

$$\text{CRC} = 11000001$$

这样，将 80 比特的加密数据和 8 比特的 CRC 以及 8 比特全 0 的编码器尾串联，得到

1110100100 0011000011 1000100111 0011110111 100110110
0100000101 0101010011 0101111010 1100000100 000000

将加密后的密文输入(3, 1, 8)卷积编码器，编码器的输出是

c_0：1100101110 1110011111 0100110001 1101100101 1111010000 0110000011
1001101101 1100000010 1001101100 101111

c_1：1000111100 0000000111 0100000000 0101111000 1100010000 1111111100
0000111010 0101101000 1101101011 110011

结果，得到最后的加密密文为

11110000010001100101101101000001 10110010100000001001001111110110
00011100100110010000000000001100 10011100111111001001110000110O
11111110110100111000100100100 1 01111111101001101101001110110 1
10100000001001100111101000101 01 10111100101001100101100110100 1
11101000111111100111010101001 0 11001010110011011 1

例 8.16　考虑每帧96比特的已串联数据的加密问题，即在 4 800 b/s 发送速率下，信息数据假设为

1010110011 1001101111 0010010100 0100110011

0011001000 1110010110 1001100110 0110011110

计算出的 CRC 比特为

10111001

这样，通过加上 8 比特编码器尾比特（额外的全 0 比特）就可以简单地算出 4 800 b/s 下 96 比特的帧数据，如下：

1010110011 1001101111 0010010100 0100110011 0011001000

1110010110 1001100110 0110011110 1011100100 000000

这就是要用密钥序列加密的数据序列，其中密钥序列是从非线性组合器中生成的，在 8.13.1 节中已经讨论论过。密钥序列为

0100010111 1010101100 1010110011 0111000100 1010100110

1010010011 1100110101 0011100100 1100000010 100110

将信息帧序列和密钥序列进行异或运算，得到密文序列为：

1110100100 0011000011 1000100111 0011110111 100110110

0100000101 0101010011 0101111010 0101110110 100110

这就是加密后的帧序列，输入到（3，1，8）卷积编码器。

移动台先对反向业务信道上发送的加密后的数据序列进行卷积编码，然后再进行块交织。每输入一个比特编码器产生三个码符号（c_0，c_1，c_2），计算出的三个码符号如下：

c_0 : 1100101110 1110011111 0100110001 1101100101 1111010000 0110000011
c_1 : 1000111100 0000000111 0100000000 0101111000 1100010000 1111111100
c_2 : 1010000001 1010000100 0111000010 0111001010 1111101111 1110110111

所以，最后的加密码符号为

11110000010001100101101101000011 01100101000000100100111110110
00011100100110010000000000001100 1001110011111001001110000 1100
11111110110100111000100100100 1 0111111110100110110100111 01101
10100000001001100111101000101 01 1011110010100110010110011 01001
00000110011000110101001001101 0 011010011111100010

至此，外部加密的情况已经讨论，接下来讨论消息信息的内部加密。在反向业务信道上发送的数据帧封装成 20 ms 一组，发送数据速率为 4 800 b/s 时，每帧的长度为 96 比特，其中包含 80 比特的信息、8 比特的 CRC 和 8 比特的编码器尾比特。

内部加密方案如图 8.22(b)所示，首先，对 96 比特的消息帧进行(3，1，8)卷积编码，从编码器输出的码符号是输入的三倍长，这些码符号用密钥序列进行加密，其中密钥序列是在 8.13.1 节中讨论的非线性组合器产生的。

通过下面的例子来说明内部加密方法。

例 8.17 在图8.22(b)的基础上，本例考虑在 4.8 kb/s 速率下的内部加密问题。

假设信息序列为：

1010110011 1001101111 0010010100 0100110011
0011001000 1110010110 1001100110 0110011110

计算出的 CRC 比特为：10111001

编码器尾比特为：00000000

这样，将信息数据与 CRC 比特和尾比特串联后的表示如下：

1010110011 1001101111 0010010100 0100110011 0011001000
1110010110 1001100110 0110011110 1011100100 000000

如图 3.4 所示将这个序列输入(3，1，8)卷积编码器，对应于每个输入比特计算出产生的码符号如下：

c_0 :1001010001 1000100001 0111110000 1001100111 1001100011 1111110111
1011100111 0011001010 1100010011 011011

c_1 :1110010000 1100100001 0001000000 0001101100 1010101100 0101110010
1101010001 0101010001 0100101001 010111

c_2 :1101010001 1001111110 0110100011 0100111110 1100110100 0010000111
0101000101 1001100011 0001111010 110101

串联从模 2 加法器输出的所有三个符号（c_0，c_1，c_2），得到一个单一的输出序列，作为符号重复器和块交织器的输入。因为编码率为 1/3，所以输出的码符号是 288 比特，如下所示：

1110110101 0100011100 0000000101 1110100000 0111100100
1001001110 0001011011 1010110000 0000001001 1000010001
1011100101 1111101100 1110010101 0011100101 0011100100
1001101011 1011011000 0101111101 1100111001 1110001000
0101100111 0010101001 1100101010 0000101011 1001100000
0101110101 1000101110 0011111000 11110111

接着，从密钥发生器产生的经非线性组合器过滤得到的密钥流（仅 288 比特）为：

0100010111 1010101100 1010110011 0111000100 1010100110
1010010011 1100110101 0011100100 1100000010 1001100101
0001010110 0001110010 1011111101 0001111101 0101000110
0100001110 0100010010 1110100101 1110010001 0101000010
1010110100 1100001101 0011111110 0001111110 0010101111
1000101000 0100101000 0110000001 1110000

最后，将输出的码符号与密钥流进行异或运算，得到加密后的数据序列如下所示：

1010100010 1110110000 1010110110 1001100100 1101000010
0011011101 1101101110 1001010100 1100001011 0001110100
1010110011 1110011110 0101101000 0010011000 0110100010
1101100101 1111001010 1011011000 0010101000 1011001010
1111010011 1110100100 1111010100 0001010101 1011001111
1101011101 1100000110 0101111001 00010111

第 9 章　反向 W-CDMA 信道

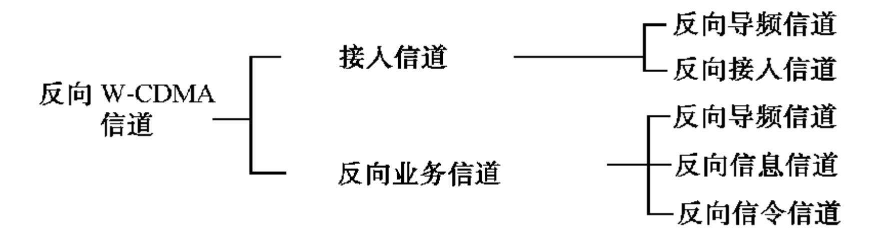
宽带 CDMA(W-CDMA)个人通信服务(PCS)标准具有如下特性:

- 用统一的服务来完成高层和低层的应用。
- 通过自适应差分脉冲编码调制(ADPCM)模式完成通话质量服务。
- 对于每小区 5 MHz 带宽可提供 128 个信道的系统容量;10 MHz 带宽可提供 256 个信道系统容量;15 MHz 带宽可提供 384 个信道系统容量,并且具有消除干扰和语音激活检测功能。消除干扰系统(ICS)可以消除小区内的自干扰并且能够增加已经建立的可用信道数。
- 通过宽带调制,高速音频数据传输服务可达到 64 kb/s。ISDN 的数据传输速率可超过 144 kb/s。

基本的 W-CDMA 系统每一载频为 5 MHz。通过(2, 1, 8)卷积编码器可以降低比特传输错误。为了达到在前向链路中区分(或信道化)不同信道的目的,每个信道都分配了一个惟一的正交码。通过使用直接序列扩频,可以使码片速率达到 4.096 Mc/s。每个信道在经过 BPSK 扩展方式调制后还要经过 QPSK 调制。然而当 W-CDMA 系统每一载频为 10 MHz 或15 MHz时,相应的扩频速率可以更高。

反向 W-CDMA 信道是指移动台到基站的通信链路。反向链路包括一个接入信道和一个业务信道,每个信道又包括一个反向导频信道,反向业务信道还包括一个信令信道。每个移动台发送一个反向导频信号,该信号与基站发送的导频信号是同步的。

反向 CDMA 信道由接入信道和反向业务信道组成。接入信道由以下两类信道组成:(1)反向导频信道,(2)反向接入信道;反向业务信道由以下 3 类信道组成:(1)反向导频信道,(2)反向信息信道,(3)反向信令信道。



通过使用直接序列 CDMA 技术,这些信道可以共享指定的 CDMA 频率。每一个接入信道和反向业务信道都通过一个远端用户长码序列来识别。

在第九章和第十章中将介绍 W-CDMA PCS 标准。它包括通过提供 5 MHz 的带宽使每小区具有 128 个信道的系统容量和通过宽带调制使高速话音频带的数据传输服务可达到64 kb/s两个方面。数据传送速率分别是 64 kb/s,32 kb/s 和 16 kb/s。其中,64 kb/s 的传送速率可以支持 G4 传真和视频传送;32 kb/s 的数据传送可以支持 32 kb/s ADPCM;16 kb/s 速率可以支持16 kb/s的 LD-CELP。码激励线性预测编码(CELP)是一种语音编码算法。CELP 使用了随机激发器、长时预测滤波器和短期格式预测滤波器。

9.1　反向业务信道

反向业务信道支持在针对反向信息信道的 64 kb/s,32 kb/s 和 16 kb/s 以及针对反向信令信道的 8 kb/s,4 kb/s,2 kb/s 下的可变数据速率。反向信息信道和反向信令信道每帧 5 ms。反向业务信道用于通信过程中由移动台向基站传输语音、数据和必要的信令信息。

反向 W-CDMA 信道中反向业务信道的结构如图 9.1 所示。所有在反向 W-CDMA 信道中传送的数据在被传送之前,都要通过直接序列扩展进行调制、卷积编码、交织。

反向信息信道和反向信令信道的调制参数如表 9.1 和 9.2 所示。

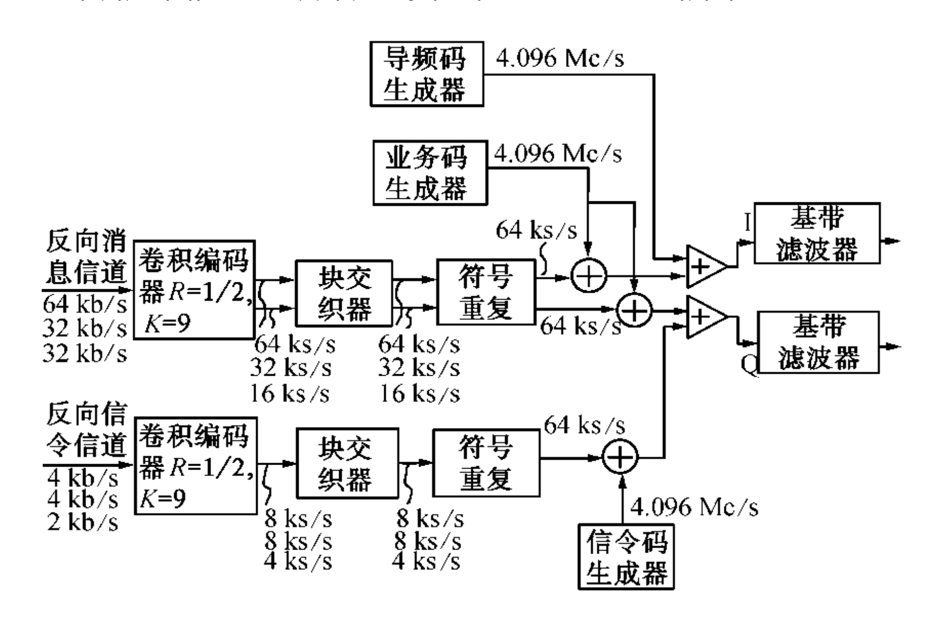


图 9.1　反向 W-CDMA 的反向业务信道结构[JTC(空中)/95 之后]

表 9.1　反向信息信道调制参数

参数	速率(b/s)			单位
	64 000	**32 000**	**16 000**	
PN 码率	4.096	4.096	4.096	Mc/s
码率	1/2	1/2	1/2	比特/符号
符号重复	1	2	4	
符号速率	64 000	64 000	64 000	s/s
PN 码片/符号	64	64	64	PN 码片/符号
PN 码片/符号	64	128	256	PN 码片/比特

表 9.2　反向信令信道调制参数

参数	速率(b/s)		单位
	4 000	**2 000**	
PN 码率	4.096	4.096	Mc/s
码率	1/2	1/2	比特/符号
符号重复	8	16	
符号速率	64 000	64 000	s/s
PN 码片/符号	64	64	PN 码片/符号
PN 码片/比特	512	1 024	PN 码片/比特

9.2 反向信息信道(RIC)

每帧 5 ms 的反向信息信道可以在 64 kb/s,32 kb/s 及 16 kb/s 等不同速率下传送,分别包含 320 比特,160 比特和 80 比特。

9.2.1 RIC 卷积编码

移动台在对将要被传送的数据交织前,要先进行卷积。卷积编码率为 $R=1/2$,约束长度为 $K=9$。

生成多项式分别是:

$$g_0(x)=1+x+x^2+x^3+x^5+x^7+x^8$$

$$g_1(x)=1+x^2+x^3+x^4+x^8$$

对应每个输入编码器的数据比特,(2,1,8)卷积码产生两个编码符号 c_0 和 c_1。在信息信道和信令信道中,分别由 g_0 和 g_1 所编的码符 c_0 和 c_1 先后输出。编码器的初始状态为全零,初始化以后的第一个输出码符应是 g_0 所编的码符 c_0。如图 9.2 所示,为该编码器 8 级移位寄存器和两个模 2 加法器的组成。

例 9.1 假定反向业务信道的反向信息信道速率为 16 kb/s,在此速率下,反向信息信道每帧有 80 比特,让我们假定在信道中传送的序列为:

1100011100 0010100100 1100110111 0101011001

1100110010 1001110101 1001111100 0110011110

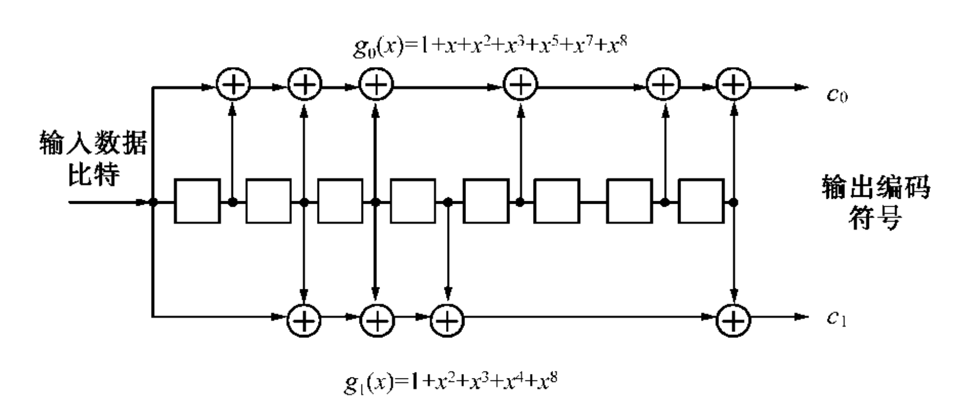


图 9.2 (2,1,8)卷积编码器

编码器初始状态为全零,如图 9.2 所示,编码器对于每个输入的数据 d 都产生两个编码符号 c_0 和 c_1。如下表:

d	c_0	c_1
1	1	1
1	0	1
0	0	1
0	0	0
0	1	0
1	0	0
·	·	·
·	·	·

·	·	·

因此，编码器经过运算输出的相对应的符号序列为：

c_0：1000101011 0111011111 1100101011 0010000101
　0100010001 1110010001 1011011011 1010011011

c_1：1110001010 0111100011 1000111100 0010101100
　0100001100 0111001011 0110011101 0100100010

9.2.2　RIC 块交织

在调制和传送前，移动台交织所有在反向业务信道上的编码符号。常用的交织跨度为5 ms，10 ms或20 ms，这些交织跨度是可选的，并且只用于反向业务信道中的反向信息信道。反向信息信道中的编码符号根据不同情况进行交织，具体如下，比如针对交织跨度为5 ms的情况，交织器是一个 20 行、16 列（320 个元素）的阵列，在反向业务信道的反向信息信道中编码符号按列写入交织器，填满整个 20×16 的矩阵；在反向业务信道的信令信道中编码符号按列写入交织器，填满整个 8×5 的矩阵（如表 9.5）。

针对交织跨度为5 ms的情况，表 9.3 中给出反向业务信道的反向信息 I 信道中编码符号进入交织输入矩阵时写操作的顺序；表 9.4 给出反向业务信道的反向 Q 信道的写操作顺序。

表 9.3　反向信息和接入信道（I）的交织器输入（写入速率为 16 kb/s）

跨度为 5 ms 的交织器							
1	11	21	31	41	51	61	71
2	12	22	32	42	52	62	72
3	13	23	33	43	53	63	73
4	14	24	34	44	54	64	74
5	15	25	35	45	55	65	75
6	16	26	36	46	56	66	76
7	17	27	37	47	57	67	77
8	18	28	38	48	58	68	78
9	19	29	39	49	59	69	79
10	20	30	40	50	60	70	80

表 9.4　反向信息和接入信道（Q）交织器的输入（写入速率为 16 kb/s）

跨度为 5 ms 的交织器							
36	46	56	66	76	6	16	26
37	47	57	67	77	7	17	27
38	48	58	68	78	8	18	28
39	49	59	69	79	9	19	29
40	50	60	70	80	10	20	30
41	51	61	71	1	11	21	31
42	52	62	72	2	12	22	32
43	53	63	73	3	13	23	33
44	54	64	74	4	14	24	34
45	55	65	75	5	15	25	35

表 9.5 和表 9.6 给出信令信道的编码符号分别以 4 比特（8×5）和 2 比特（4×5）的速率输入交

织矩阵时写操作的顺序。反向信息信道和信令信道的编码符号从交织器中按行输出。

表 9.5　信令信道交织器输出(写入速率为 **4 kb/s**)

1	9	17	25	33
2	10	18	26	34
3	11	19	27	35
4	12	20	28	36
5	13	21	29	37
6	14	22	30	38
7	15	23	31	39
8	16	24	32	40

表 9.6　信令信道交织器输出(写入速率为 **2 kb/s**)

1	5	9	13	17
2	6	10	14	18
3	9	11	15	19
4	8	12	16	20

例 9.2　假定 80 比特的反向业务信道的反向信息信道速率为 16 kb/s,如果 I 信道卷积编码器(2,1,8)的符号序列 c_0 如下所示,利用表 9.3 按列输入交织编码器:

输入符号 c_0
(按列写入)
⇩
10100111
01101100
01010111
01000010
10100000　⇨　**输出符号 c_0** **(按行读出)**
01001111
11100011
01010000
11100011
11111111

则从交织编码器按行输出的符号 c_0 序列为:

10100111 01101100 01010111 01000010 10100000

01001111 11100011 01010000 11100011 11111111

为了便于使用表 9.4,Q 信道的符号序列 c_1 将被按照下面的习惯安排:

比特位	编码符号
1 ~ 10	1110001010
11 ~ 20	0111100011
21 ~ 30	1000111100
31 ~ 40	0010101100
41 ~ 50	0100001100
51 ~ 60	0111101011
61 ~ 70	0110011101
71 ~ 80	0100100010

以Q信道的符号序列为输入的交织器输出结果可以依照表9.4通过 c_1 序列写操作的顺序得到。因而,Q信道 c_1 序列通过交织器的结果逐行读出如下:

36,46,56,66,76,6,16,26　→　00010001
·　11110101
·　11010001
·　00101110
　00110010
41,51,61,71,1,11,21,31　→　00001010
·　11111100
·　01101101
·　01000100
45,55,65,75,5,15,25,35　→　00010111

最后,Q信道经过交织后的符号序列,即交织器的输出结果展开如下:

00010001 11110101 11010001 00101110 00110010
00001010 11111100 01101101 01000100 00010111

9.2.3 RIC 符号重复

交织之后,反向业务信道的符号根据速率不同进行重复。在扰码前,每个64 kb/s速率的符号不重复;32 kb/s速率的符号重复1次(每个码元出现两次),每个16 kb/s速率的符号重复3次(每个码元出现4次)。

例9.3　速率为16 kb/s的符号重复:

I信道交织后的符号序列如下,

10100111　01101100　01010111　01000010　10100000
01001111　11100011　01010000　11100011　11111111

每个符号重复3次,则符号重复的输出结果为:

1111　0000　1111　0000　0000　1111　1111　1111
0000　1111　1111　0000　1111　1111　0000　0000
0000　1111　0000　1111　0000　1111　1111　1111
0000　1111　0000　0000　0000　0000　1111　0000
1111　0000　1111　0000　0000　0000　0000　0000
0000　1111　0000　0000　1111　1111　1111　1111
1111　1111　1111　0000　0000　0000　1111　1111
0000　1111　0000　1111　0000　0000　0000　0000
1111　1111　1111　0000　0000　1111　1111　1111
1111　1111　1111　1111　1111　1111　1111　1111

Q信道交织后的符号序列如下,

00010001　11110101　11010001　00101110　00110010
00001010　11111100　01101101　01000100　00010111

每个符号重复3次,则符号重复的输出结果为:

0000　0000　0000　1111　0000　0000　0000　1111
1111　1111　1111　1111　0000　1111　0000　1111
1111　1111　0000　1111　0000　0000　0000　1111
0000　0000　1111　0000　1111　1111　1111　0000
0000　0000　1111　1111　0000　0000　1111　0000
0000　0000　0000　0000　1111　0000　1111　0000
1111　1111　1111　1111　1111　1111　0000　0000
0000　1111　1111　0000　1111　1111　0000　1111
0000　1111　0000　0000　0000　1111　0000　0000
0000　0000　0000　1111　0000　1111　1111　1111

9.2.4 RIC 直接序列扩展

在传输之前，反向业务信道的反向信息信道和信令信道被导频码序列、业务码 I 序列、业务码 Q 序列和信令码序列直接扩展。如图 9.3 所示，经过 81 920 个码片的一个周期，通过长码生成器和适当的 Walsh 函数的模 2 加运算，便可由 1 个码序列生成以上的码序列。在图9.3(b)中，编号为 0 的 Walsh 函数被指定参与运算生成导频码；编号为 1 的 Walsh 函数被指定参与运算生成业务码 I 和业务码 Q，以及接入码；编号为 2 的 Walsh 函数被指定参与运算生成信令码。每个码序列在以 20 ms 为 1 周期的 81 920 个码片中产生。作为长码使用的 PN 码序列设计如图 9.3(a)所示。

长码初始值随着寻呼信道的信息被基站发送，这个长码是具有 $2^{23}-1$ 个码片周期的截断了的线性码，并且满足下面生成多项式的线性递归的要求：

$$p(x)=1+x+x^2+x^{22}+x^{32}$$

以 4.096 Mc/s 为固定码率的 Walsh 函数对反向业务信道和接入信道的所有码信道(反向导频信道、反向信息信道和反向信令信道)正交信道化[见图 9.3(b)所示]。上面使用的 3 个码长为 64 的 Walsh 函数由表 9.7 定义给出。

表 9.7　反向链路的 **Walsh** 函数

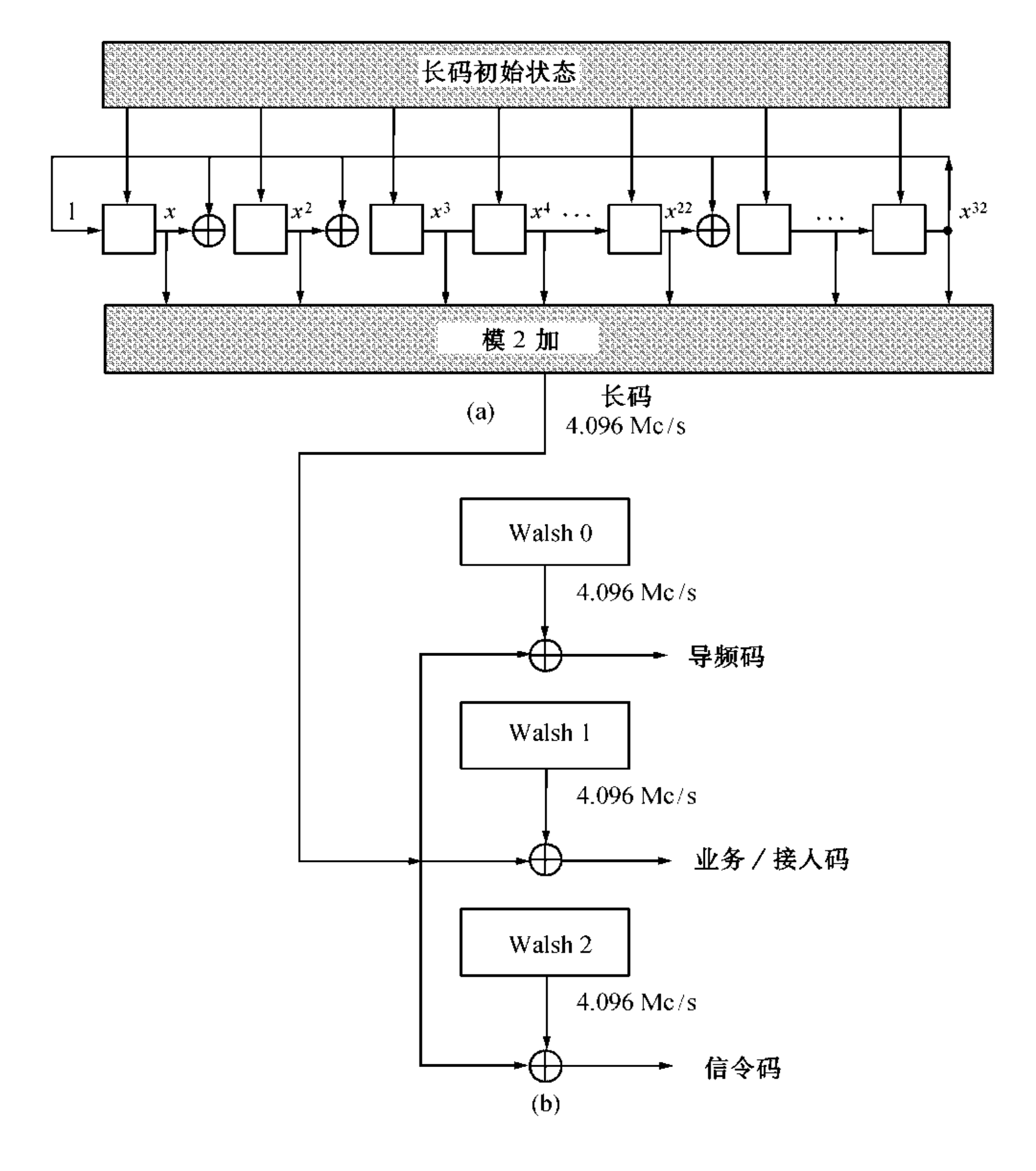
Walsh 函数码片

Walsh 编号	0123	4567	11 8901	1111 2345	1111 6789	2222 0123	2222 4567	2233 8901	3333 2345	3333 6789	4444 0123	4444 4567	4455 8901	5555 2345	5555 6789	6666 0123
0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
1	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101
2	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011

采用 64 阶 Walsh 函数区分逻辑信道，分别用 W0，W1，…，W63 表示，如表 9.7，第 1 行 W0 为全零，作为每一帧的起始传输，即零偏置，帧长 5 ms。

直接序列扩展后，每个二进制数(0 或 1)将映像到相位上：0 对应 $+\pi$ 相位，1 对应 0 相位。

例 9.4　参照图 9.3(a)，PN 码生成器的抽头多项式为 $g(x)=1+x+x^2+x^{22}+x^{32}$，它的向量形式为 1110000000 0000000000 0010000000 001；初始值向量假定为 1001001100 1010011011 0101111000 11，利用这两个向量生成的长码序列如下表 9.8 所示。



(a)生成长码的 PN 码生成器　(b)生成长码的 PN 序列生成器

图 9.3　码信道间的正交信道化

表 9.8　利用 **PN** 码生成器生成长码

偏移编号	寄存器内容	∑'1's	长码比特
0	10010011001010011011010111100011	17	1
1	10101001100101001101100011110001	16	0
2	10110100110010100110111001111000	17	1
3	01011010011001010011011100111100	17	1
4	00101101001100101001101110011110	17	1
5	00010110100110010100110111001111	17	1
6	11101011010011001010010011100111	18	0
7	10010101101001100101000001110011	15	1
8	10101010110100110010101000111001	16	0
9	10110101011010011001011100011100	17	1
10	01011010101101001100101110001110	17	1

因此生成的长码序列如下：

1011110101110100 0010111010110010 1001111011101011 0111001111000001
0010100011011010 0110110111011100 1010110100101100 0100101000000101
1100001010110101 0101101101001111 1001111011101101 0011101110100000
0011000010001111 1000000110000000 1111001110010010 0110000000011010
1101110001001000 0011001001001000 1000011111011001 0110100110100001
1111111101110001 1100000000010101 1111100100101011 0000101010101011
0100010100000000 0111010101001101 1000100001100011 0101001000110100
0011010000100101 1011011111100001 0111010010100101 0010111011101101
1111010000000110 1100100100110011 1000000011110000 0100001110001010
0110000001001001 11011000

例 9.5 如图 9.3(b)中所示，通过长码序列和适当的 Walsh 函数的模 2 加可以得到导频码、业务/接入码和信令码。利用 3 个 64 位长 Walsh 函数(见表 9.7)可以用 1 个周期的81 920个码片生成以上的码序列。

1. 导频码的生成

编号为 0 的 Walsh 函数 W0 被指定与长码运算生成导频码。

$$导频码 = 长码 \oplus W0$$

长码： 1011110101110100 0010111010110010 1001111011101011 011100111100001

$\oplus$

Walsh 0： 0000000000000000 0000000000000000 0000000000000000 000000000000000

导频码： 1011110101110100 0010111010110010 1001111011101011 011100111100001

这个计算的示例只是长码序列的第 1 行。

因此长码序列和 Walsh 函数 W0 通过模 2 加法生成的导频码为：

1011110101110100 0010111010110010 1001111011101011 0111001111000001
0010100011011010 0110110111011100 1010110100101100 0100101000000101
1100001010110101 0101101101001111 1001111011101101 0011101110100000
0011000010001111 1000000110000000 1111001110010010 0110000000011010
1101110001001000 0011001001001000 1000011111011001 0110100110100001
1111111101110001 1100000000010101 1111100100101011 0000101010101011
0100010100000000 0111010101001101 1000100001100011 0101001000110100
0011010000100101 1011011111100001 0111010010100101 0010111011101101

可以看出，导频码只是长码序列的复制。

2. 业务/接入码的生成

编号为 1 的 Walsh 函数 W1 被指定与长码运算生成业务/接入码。

$$业务/接入码 = 长码 \oplus W1$$

长码： 1011110101110100 0010111010110010 1001111011101011 0111001111000001

$\oplus$

Walsh 1： 0101010101010101 0101010101010101 0101010101010101 0101010101010101

业务/接入码：1110100000100001 0111101111100111 1100101110111110 0010011010010100

长码序列和 Walsh 函数 W1 通过模 2 加法生成的业务/接入码为：

1110100000100001 0111101111100111 1100101110111110 0010011010010100
0111110110001111 0011100010001001 1111100001111001 0001111101010000
1001011111100000 0000111000011010 1100101110111000 0110111011110101
0110010111011010 1101010011010101 1010011011000111 0011010101001111
1000100100011101 0110011100011101 1101001010001100 0011110011110100
1010101000100100 1001010101000000 1010110001111110 0101111111111110
0001000001010101 0010000000011000 1101110100110110 0000011101100001
0110000101110000 1110001010110100 0010000111110000 0111101110111000

3. 信令码生成

编号为 2 的 Walsh 函数 W2 被指定与长码运算生成信令码。

$$信令码 = 长码 \oplus W2$$

长码：　1011110101110100 0010111010110010 1001111011101011 0111001111000001

⊕

Walsh 2：0011001100110011 0011001100110011 0011001100110011 0011001100110011

信令码：　1000111001000111 0001110110000001 1010110111011000 0100000011110010

长码序列和 Walsh 函数 W2 通过模 2 加法生成的信令码为：

1000111001000111 0001110110000001 1010110111011000 0100000011110010
0001101111101001 0101111011101111 1001111000011111 0111100100110110
1111000110000110 0110100001111100 1010110111011110 0000100010010011
0000001110111100 1011001010110011 1100000010100001 0101001100101001
1110111101111011 0000000101111011 1011010011101010 0101101010010010
1100110001000010 1111001100100110 1100101000011000 0011100110011000
0111011000110011 0100011001111110 1011101101010000 0110000100000111
0000011100010110 1000010011010010 0100011110010110 0001110111011110

如图 9.1 所示在传输前，反向信息信道和信令信道被业务码 *I* 序列、业务码 *Q* 序列及信令码序列在发送前直接序列扩展。

反向 W-CDMA 信道中数据传输的帧长为5 ms。反向信息信道中每帧传输速度为64 kb/s，32 kb/s和16 kb/s，分别相应地包含了 320，160 和 80 比特。

例 9.6　在5 ms的帧中，符号重复器输出速率为16 kb/s，包括了 320 比特，业务码序列速率为 4.096 Mc/s，包括了 64 × 320 比特。因此，这两个序列的对应比特率为 1 到 64。

1. 反向 I 信道信息被业务码 *I* 序列直接序列扩展的结果如下：

I 信道符号重复输出(仅第一行)是 1111 0000 1111 0000 0000 1111 1111 1111。

业务码 *I* 序列是：

1110100000100001 0111101111100111 1100101110111110 0010011010010100
0111110110001111 0011100010001001 1111100001111001 0001111101010000
1001011111100000 0000111000011010 1100101110111000 0110111011110101

0110010111011010 1101010011010101 1010011011000111 0011010101001111

1000100100011101 0110011100011101 1101001010001100 0011110011110100

1010101000100100 1001010101000000 1010110001111110 0101111111111110

0001000001010101 0010000000011000 1101110100110110 0000011101100001

0110000101110000 1110001010110100 0010000111110000 0111101110111000

由于 I 信道符号重复序列中的 1 个符号对应着业务码 I 序列的 64 个比特,因而两个序列模 2 加的结果如下:

1→ 0001011111011110 1000010000011000 0011010001000001 1101100101101011

1→ 1000001001110000 1100011101110110 0000011110000110 1110000010101111

1→ 0110100000011111 1111000111100101 0011010001000111 1001000100001010

1→ 1001101000100101 0010101100101010 0101100100111000 1100101010110000

0→ 1000100100011101 0110011100011101 1101001010001100 0011110011110100

0→ 1010101000100100 1001010101000000 1010110001111110 0101111111111110

0→ 0001000001010101 0010000000011000 1101110100110110 0000011101100001

0→ 0110000101110000 1110001010110100 0010000111110000 0111101110111000

2. 另一方面,Q 信道符号重复序列(仅第 1 行)是由:0000 0000 0000 1111 0000 0000 0000 1111 重复,Q 信道重复序列与业务码 Q 序列的模 2 加结果为:

0→ 1110100000100001 0111101111100111 1100101110111110 0010011010010100

0→ 0111110110001111 0011100010001001 1111100001111001 0001111101010000

0→ 1001011111100000 0000111000011010 1100101110111000 0110111011110101

0→ 0110010111011010 1101010011010101 1010011011000111 0011010101001111

0→ 1000100100011101 0110011100011101 1101001010001100 0011110011110100

0→ 1010101000100100 1001010101000000 1010110001111110 0101111111111110

0→ 0001000001010101 0010000000011000 1101110100110110 0000011101100001

0→ 0110000101110000 1110001010110100 0010000111110000 0111101110111000

9.3 反向信令信道(RSC)

针对反向信令信道(RSC),反向业务信道支持 2 种速率4 kb/s和2 kb/s。信令信道每帧的跨度为5 ms。反向业务信道的反向信令信道如图 9.1 所示。所有反向业务信道中的数据在传输前都要经过卷积编码、交织和直接序列扩展调制的过程。

在传输速率为4 kb/s的反向信令信道每帧有 20 比特,其中 2 比特为不连续传输标识,18 比特为信令信息。而以2 kb/s为传输速率的反向信道每帧包含 10 比特。

反向信令信道的调制参数如表 9.2。

9.3.1 RSC 卷积编码

反向信令信道的数据在发射前要进行卷积编码。码率为 1/2,约束长度 $K = 9$,生成器抽头多项式为:

$$g_0(x) = 1 + x + x^2 + x^3 + x^5 + x^7 + x^8$$
$$g_1(x) = 1 + x + x^2 + x^3 + x^4 + x^8$$

如图9.2所示，反向信令信道用(2，1，8)卷积编码器进行卷积编码。该编码器由 $K-1=8$ 级移位器和两个模2加法器组成。在反向信令信道中，输入编码器的每个比特都会生成2个编码符号。第一个编码符号 c_0 由抽头多项式 $g_0(x)$ 生成，第二个编码符号 c_1 则由抽头多项式 $g_1(x)$ 生成。(c_0, c_1) 成对输出并串联成一个符号序列。

例9.7　设反向业务信道的反向信令信道中数据速率为2 kb/s，在该数据速率下反向信令信道帧是1101000101(10比特)。编码器初始状态为全零，对于每个输入的数据所计算出的两个输出符号 c_0 和 c_1 如表9.9。

表9.9　反向信令信道卷积编码器输出符号(数据速率2 kb/s)

偏移编号	编码器内容	(c_0, c_1)输出对
1	100000000	(1,1)
2	110000000	(0,1)
3	011000000	(0,1)
4	101100000	(1,1)
5	010110000	(0,0)
6	001011000	(0,0)
7	000101100	(0,1)
8	100010110	(0,0)
9	010001011	(0,1)
10	101000101	(1,1)

因此，编码器输出符号为：

$$c_0 = (1001000001)$$
$$c_1 = (1111001011)$$

卷积编码器生成的符号序列为：

$$c = (1101011100\ 0001000111)$$

例9.8　设反向业务信道的反向信令信道中数据速率为4 kb/s，在该速率下反向信令信道帧是11010001011011010011(20比特)。编码器初始状态为全零，经计算编码器输出符号 c_0 和 c_1 为：

$$c_0 = (1001000001\ 0101011001)$$
$$c_1 = (1111001011\ 0111000110)$$

则卷积编码器生成的符号序列为：

$$c = (1101011100\ 0001000111\ 0011011100\ 1010010110)$$

9.3.2　RSC块交织

在反向业务信道的反向信令信道中的编码符号，在被调制和发射之前都会经过交织。数据速率为4 kb/s的RTC反向信令信道编码符号将按列输入交织器，填满8×5的矩阵(见表9.5)；而数据速率为2 kb/s的编码符号将填满4×5的矩阵(见表9.6)。反向信令信道编码符号将按行从交织器中输出。

例9.9　设反向业务信道的反向信令信道中数据速率为2 kb/s，从例9.7中我们得到卷积编码器输出序列 $c=(11010111000001000111)$，按照下面的规则将 c 按列写入交织器，填满4×5的矩阵，则

我们可得到：

按列输入交织器
⇩
10000
11011　　⇨ 按行从交织器输出
01001
11001

经过 4×5 的矩阵计算，交织器按行输出的结果为：10000 11011 01001 11001

例 9.10　设 RTC 信令信道数据速率为4 kb/s，从例 9.8 中我们得到卷积编码器输出序列 c =（1101011100 0001000111 0011011100 1010010110）

将 c 的所有码符号按列输入交织器，填满 8×5 的矩阵，则我们可得到：

按列输入交织器
⇩
10001
10110
00110
10111　　⇨ 按行从交织器输出
00000
11001
10111
10100

交织器输出结果为：

10001 10110 00110 10111

00000 11001 10111 10100

9.3.3 RSC 符号重复

交织之后，RTC 反向信令信道的符号根据速率不同进行重复。4 kb/s速率的符号重复 7 次（每个码元连续出现 8 次），2 kb/s速率的符号重复 15 次（每个码元连续出现 16 次）。

例 9.11　在例 9.9 中，交织器输出数据速率为2 kb/s的码序列为 10000 11011 01001 11001，每个符号重复 15 次结果如下：

1111111111111111 0000000000000000 0000000000000000 0000000000000000

0000000000000000 1111111111111111 1111111111111111 0000000000000000

1111111111111111 1111111111111111 0000000000000000 1111111111111111

0000000000000000 0000000000000000 1111111111111111 1111111111111111

1111111111111111 0000000000000000 0000000000000000 1111111111111111

例 9.12　交织器输出数据速率为4 kb/s的码序列为：

10001 10110 00110 10111

00000 11001 10111 10100

由于交织器输出数据速率为8 ks/s，每个码率为4 kb/s的符号重复 7 次，则符号重复器输出码率为64 ks/s，即：

8×8（每个码元连续出现 8 次） = 64 ks/s。

因而，符号重复器输出如下：

11111111 00000000 00000000 00000000 11111111

11111111 00000000 11111111 11111111 00000000

00000000 00000000 11111111 11111111 00000000

11111111 00000000 11111111 11111111 11111111

00000000 00000000 00000000 00000000 00000000

11111111 11111111 00000000 00000000 11111111

11111111 00000000 11111111 11111111 11111111

11111111 00000000 11111111 00000000 00000000

9.3.4　RSC 直接序列扩展

反向业务信道的反向信令信道被信令码序列直接序列扩展。如图 9.3 所示，信令码序列由长码和编号为 2 的 Walsh 函数模 2 加得到。由于调制符号是通过 Walsh 函数得到的 64 个正交波形，所以这一调制过程称为正交调制。信令码序列由81 920个码片在 1 个周期（20 ms）产生，从图 9.3（a）中可知，PN 码用来生成长码。

利用下面的公式，在 9.2.4-(3)中已经计算出信令码序列：

$$信令码 = 长码 \oplus W2$$

则 9.2.4-(3)中生成的信令码如下：

1000111001000111 0001110110000001 1010110111011000 0100000011110010

0001101111101001 0101110111101111 1001111000011111 0111100100110110

1111000110000110 0110100001111100 1010110111011110 0000100010010011

0000001110111100 1011001010110011 1100000010100001 0101001100101001

1110111101111011 0000000101111011 1011010011101010 0101101010010010

1100110001000010 1111001100100110 1100101000011000 0011100110011000

0111011000110011 0100011001111110 1011101101010000 0110000100000111

0000011100010110 1000010011010010 0100011110010110 0001110111011110

1100011100110101 1111101000000000 1011001111000011 0111000010111001

0101001101111010 1110101111110010 0010001010001010 0101011100101101

0100000010000001 1110001111011100 0101001011110111 0111000010101111

1111000101011110 0110001010110001 0001110010111010 0110010111101000

在5 ms 时间内，RSC 符号重复器输出速率为64 kb/s的序列，每帧 320 个符号；信令码固定速率为 4.096 Mc/s，共 64 ×320 个码片。所以 PN 码片/符号比为 $4.096 \times 10^6/(64 \times 10^3) = 64$。

例 9.13　考虑 RSC 符号重复器的输出结果经过信令码序列直接序列扩展。仅选择重复器前32 个输出符号，如 1111111111111111 0000000000000000，这 32 个符号直接序列扩展的结果如下：

1→ 0111000110111000 1110001001111110 0101001000100111 1011111100001101

1→ 1110010000010110 1010000100010000 0110000111100000 1000011011001001

1→ 0000111001111001 1001011110000011 0101001000100001 1111011101101100

1→ 1111110001000011 0100110101001100 0011111101011110 1010110011010110

1→ 0001000010000100 1111111010000100 0100101100010101 1010010101101101

1→ 0011001110111101 0000110011011001 0011010111100111 1100011001100111

1→ 1000100111001100 1011100110000001 0100010010101111 1001111011111000

1→ 1111100011101001 0111101100101101 1011100001101001 1110001000100001

·

·

·

以上分析基于的数据速率为2 kb/s。

例 9.14　下面考虑数据速率为4 kb/s的情况。

RSC 符号重复器输出结果的直接序列扩展可以通过重复器输出结果与信令码的模 2 加得到。即：直接序列扩展 = 重复器输出结果 ⊕ 信令码。

利用例 9.12 中得到的符号重复器输出结果 111111111 00000000 00000000 ……与信令码序列相加，则得到直接序列扩展如下：

1→ 0111000110111000 1110001001111110 0101001000100111 1011111100001101

1→ 1110010000010110 1010000100010000 0110000111100000 1000011011001001

1→ 0000111001111001 1001011110000011 0101001000100001 1111011101101100

1→ 1111110001000011 0100110101001100 0011111101011110 1010110011010110

1→ 0001000010000100 1111111010000100 0100101100010101 1010010101101101

1→ 0011001110111101 0000110011011001 0011010111100111 1100011001100111

1→ 1000100111001100 1011100110000001 0100010010101111 1001111011111000

1→ 1111100011101001 0111101100101101 1011100001101001 1110001000100001

0→ 1100011100110101 1111101000000000 1011001111000011 0111000010111001

0→ 0101001101111010 1110101111110010 0010001010001010 0101011100101101

0→ 0100000010000001 1110001111011100 0101001011110111 0111000010101111

0→ 1111000101011110 0110001010110001 0001110010111010 0110010111101000

·

·

·

9.3.5　基带滤波器

经过直接序列扩展(DS)操作后，反向 I，Q 信息信道和反向信令信道还要经过如下滤波过程。如图 9.1 所示，直接序列扩展后，将 I，Q 信道脉冲输入到 I，Q 信道基带滤波器。滤波器的归一化频率响应 $20\log_{10}|S(f)|$ 包含在通带 $\delta_1 = \pm 1.5$ dB，$0 \leqslant f \leqslant f_p$，$f_p = 4.94$ MHz中，且阻带小于或等于 $\delta_2 = -40$ dB，$f \geqslant f_s$，$f_s = 4.94$ MHz。

9.4　反向导频信道

反向导频信道是供基站为反向信息信道、反向信令信道和反向接入信道获得信道的信息，用于

初始捕获、时间跟踪、参考相位获取。反向导频信道包括一种未调制的长码序列扩频信号(见 9.3.4)。反向导频信道功率水平是6 dB,低于反向信息信道的功率。

9.5　接入信道

接入信道包括两个信道:(1)反向导频信道,(2)反向接入信道。移动台通过接入信道与基站进行通信,并响应寻呼信道的信息。接入信道信号是经过编码、交织和调制扩频的信号。

移动台以16 kb/s的固定速率在接入信道中发送信息。接入信道 1 帧的持续时间为5 ms,传输由寻呼信道信息指定的时隙数目。

反向信道可以包含编号从 0 到 31 共 32 个支持寻呼信道的接入信道。针对相应的前向信道中的寻呼信道每个反向信道至少存在 1 个接入信道,每路接入信道与 1 个单独的寻呼信道相关。接入信道的长码存储元素由基站的寻呼信道设定。

基站用反向导频信道为反向接入信道获得确认、跟踪及相位参考信息。

每路反向接入信道的帧包括 80 比特(5 ms的帧)。反向接入信道结构如图 9.4 所示,其调制参数如表 9.10。

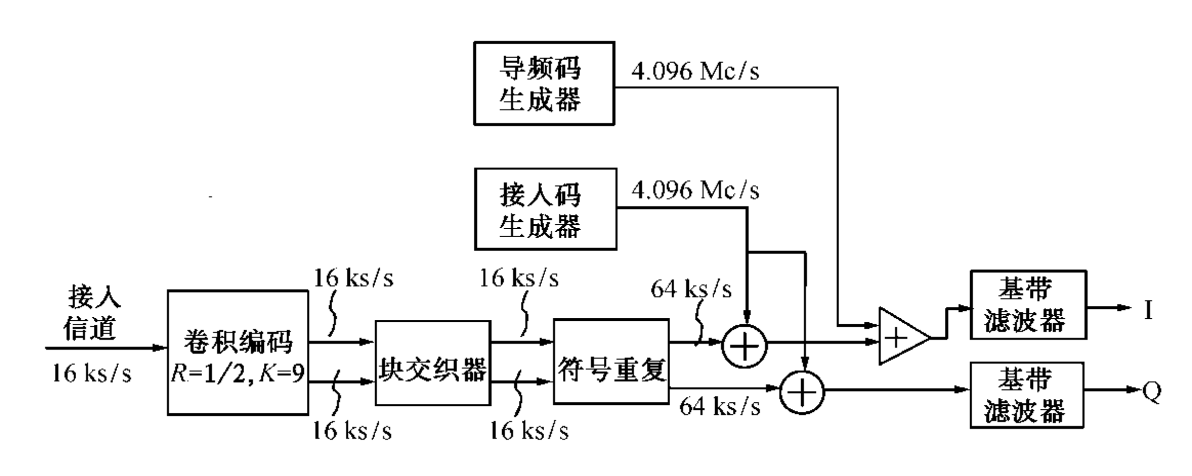


图 9.4　反向 W-CDMA 接入信道结构

表 9.10　接入信道调制参数

参　　数	数据速率(b/s)16 000	单　　位
PN 码速率	4.096	Mc/s
码率	1/2	比特/符号
符号重复	4	
符号速率	64 000	s/s
PN 码片/符号	64	PN 码片/符号
PN 码片/比特	256	PN 码片/比特

9.5.1　反向接入信道卷积编码

在节 9.2.1 中就已经提到,反向接入信道信息比特在交织前要进行卷积编码。使用(2, 1, 8)

作为卷积，卷积码率为 1/2，约束长度为 $K = 9$，生成器抽头多项式为：

$$g_0(x) = 1 + x + x^2 + x^3 + x^5 + x^7 + x^8$$

$$g_1(x) = 1 + x^2 + x^3 + x^4 + x^8$$

卷积编码器与图 9.2 中的一样，当生成反向接入信息比特时，编码器在每个5 ms帧之后都要初始化为全零状态。

例 9.15　设反向接入信道中数据速率为16 kb/s的信息序列为：

1001011100 0110100111 1101000010 0101110100

1110100100 1001111001 0100011010 1101000111

当初始状态为全零时，这 80 比特（5 ms）输入（2，1，8）卷积编码器，则编码器的输出分别为：

c_0 =（1110111010 1100110110 1110000000 0111101101

0000000010 1010110001 1101110101 1010111010）

c_1 =（1010100111 0110010001 0000010001 1110011001

0111010000 1100011101 1101001011 1000100001）

c_0 序列用于反向接入 I 信道，c_1 用于反向接入 Q 信道，这两个编码序列将输入块交织器。

9.5.2　反向接入信道块交织

移动台对所有反向接入信道中的编码符号调制和发送前要先对其交织。交织器是一个具有 5 ms交织跨度的 10 行 8 列的矩阵（80 个元素），反向接入信道的编码符号按列输入交织器，填满 10 ×8 的矩阵。

对于交织跨度为5 ms的情况，表 9.3 和表 9.4 给出了反向接入信道中编码符号输入交织器矩阵时写操作的顺序，反向接入信道的编码符号从交织器中按行输出。

例 9.16　考虑16 kb/s速率的 80 比特反向接入信道，利用表 9.3，经过（2，1，8）卷积编码后的 I 信道符号序列 c_0 按列输入交织器，形式如下：

接入 I 信道交织器输入（按列写入）

1	11	21	31	41	51	61	71		11100111
2	12	22	32	42	52	62	72		11110010
3	13	23	33	43	53	63	73		10110101
4	14	24	34	44	54	64	74		00010010
5	15	25	35	45	55	65	75	⇨	11010111
6	16	26	36	46	56	66	76		11000111
7	17	27	37	47	57	67	77		10010001
8	18	28	38	48	58	68	78		01010010
9	19	29	39	49	59	69	79		11001001
10	20	30	40	50	60	70	80		00010110

接入 I 信道交织器输出（按行读出）

因而，I 信道交织器输出如下：

11100111 11110010 10110101 00010010 11010111

11000111 10010001 01010010 11001001 00010110

例 9.17　利用表 9.4（16 kb/s速率下的写操作），经过（2，1，8）卷积编码后的 Q 信道符号序列 c_1

按列输入块交织器的形式如下：

接入Q信道交织器输入
（按列写入）

36	46	56	66	76	6	16	26		11100011	
37	47	57	67	77	7	17	27		10110000	
38	48	58	68	78	8	18	28		00100100	
39	49	59	69	79	9	19	29		00010100	接入Q信道
40	50	60	70	80	10	20	30	⇨	10111111	交织器输入
41	51	61	71	1	11	21	31		01111001	（按行读出）
42	52	62	72	2	12	22	32		11100101	
43	53	63	73	3	13	23	33		10001101	
44	54	64	74	4	14	24	34		10100000	
45	55	65	75	5	15	25	35		00011000	

因而，Q 信道交织器输出如下：

11100011 10110000 00100100 00010100 10111111

01111001 11100101 10001101 10100000 00011000

9.5.3 反向接入信道符号重复

在具有固定数据速率16 kb/s的反向接入信道中，每个交织后的符号在扰码之前，都要重复 3 次（每个码元连续出现 4 次）。

例 9.18　利用例 9.16 中获得的 I 信道交织器输出结果，重复器输出如下：

1111 1111 1111 0000 0000 1111 1111 1111

1111 1111 1111 1111 0000 0000 1111 0000

1111 0000 1111 1111 0000 1111 0000 1111

0000 0000 0000 1111 0000 0000 1111 0000

1111 1111 0000 1111 0000 1111 1111 1111

1111 1111 0000 0000 0000 1111 1111 1111

1111 0000 0000 1111 0000 0000 0000 1111

0000 1111 0000 1111 0000 0000 1111 0000

1111 1111 0000 0000 1111 0000 0000 1111

0000 0000 0000 1111 0000 1111 1111 0000

例 9.19　利用例 9.17 中获得的 Q 信道交织器输出结果，则 Q 信道重复器输出如下：

1111 1111 1111 0000 0000 0000 1111 1111

1111 0000 1111 1111 0000 0000 0000 0000

0000 0000 1111 0000 0000 1111 0000 0000

0000 0000 0000 1111 0000 1111 0000 0000

1111 0000 1111 1111 1111 1111 1111 1111

0000 1111 1111 1111 1111 0000 0000 1111

1111 1111 1111 0000 0000 1111 0000 1111

1111 0000 0000 0000 1111 1111 0000 1111

1111 0000 1111 0000 0000 0000 0000 0000

0000 0000 0000 1111 1111 0000 0000 0000

9.5.4 反向接入信道直接序列扩展

在发送之前,接入信道要被导频码序列和接入码序列(或者业务码 I 序列、业务码 Q 序列)直接序列扩展。这些码序列要由 1 个码序列经过 81 920 个码片的一个周期,通过长码生成器和适当的 Walsh 函数的模 2 加运算来生成。如图 9.3 所示,编号为 0 的 Walsh 函数被指定参与运算生成导频码;编号为 1 的 Walsh 函数被指定参与运算生成接入码。每个码序列在以20 ms为 1 周期的 81 920个码片中运算产生。

固定码速率为 4.096 Mc/s 的 Walsh 函数对反向业务信道或接入信道上的各类码信道(包括:反向导频信道、反向信息信道和反向信令信道)提供正交信道化。

如图 9.4 所示,为了得到直接序列扩展,接入 I 信道或 Q 信道上固定发送速率为64 ks/s的每个重复符号与固定码速率为4.096 Mc/s的接入码序列进行模 2 加。

参照图 9.3(a),从基站发送的寻呼信道信息中包含了长码存储元素,长码由长码存储元素通过生成器抽头多项式 $p(x) = 1 + x + x^2 + x^{22} + x^{32}$ 生成。这个长码是周期为 $2^{32} - 1$ 的截断了的线性码。

例 9.20 设长码存储元素向量为 10010011001010001 1011010111100011,生成抽头向量为 1110000000000000 0000001000000000,则从 PN 码生成器得到的长码如表 9.11 所示。

表 9.11 用 PN 码生成器生成长码

偏移编号	LFSR 内容	Σ'1's	长码比特
0	10010011001010011011010111100011	17	1
1	10101001100101001101100011110001	16	0
2	10110100110010100110111001111000	17	1
3	01011010011001010011011100111100	17	1
…	…	…	…
17	00010101101101010110100111100111	18	0
18	11101010110110101011011011110011	21	1
19	10010101011011010101100101111001	18	0
…	…	…	…
43	11110010110011001001100100110111	18	0
44	10011001011001100100111010011011	17	1
45	10101100101100110010010101001101	16	0
…	…	…	…
71	00101100001001110110111011101101	18	0
72	11110110000100111011010101110110	19	1
73	01111011000010011101101010111011	19	1
…	…	…	…
97	10001110011000000100101101110111	16	0
98	10100111001100000010011110111011	17	1
99	10110011100110000001000111011101	16	0

表 9.11 中最右侧的一列便是由 PN 码生成器生成的长码序列，具体如下：

1011110101110100 0010111010110010 1001111011101011 0111001111000001
0010100011011010 0110110111011100 1010110100101100 0100101000000101
1100001010110101 0101101101001111 1001111011101101 0011101110100000
0011000010001111 1000000110000000 1111001110010010 0110000000011010
1101110001001000 0011001001001000 1000011111011001 0110100110100001
1111111101110001 1100000000010101 1111100100101011 0000101010101011
0100010100000000 0111010101001101 1000100001100011 0101001000110100
0011010000100101 1011011111100001 0111010010100101 0010111011101101
1111010000000110 1100100100110011 1000000011110000 0100001110001010
0110000001001001 11011000

例 9.21　长码序列和编号为 0 的 Walsh 函数 W0 进行模 2 加运算，产生导频码序列，即：导频码 = 长码 ⊕ W0。由于 W0 是由 64 个全零码片构成，所以在例 9.20 中导频码恒等于长码序列。

例 9.22　考虑接入码序列的生成过程。长码序列（L. C.）和编号为 1 的 Walsh 函数 W1 进行模 2 加运算，产生接入码序列（A. C.），即 A. C. = L. C. ⊕ W1。这个码序列在 81 920 个码片（20 ms）的 1 个周期内产生。

仅考虑长码和 W1 的前 64 位，则接入码通过 A. C. = L. C. ⊕W1 生成的形式如下：

L. C.：　1011110101110100 0010111010110010 1001111011101011 0111001111000001
　⊕
W1：　　0101010101010101 0101010101010101 0101010101010101 0101010101010101
A. C.：　1110100000100001 0111101111100111 1100101110111110 0010011010010100

得到的接入码序列为：

1110100000100001 0111101111100111 1100101110111110 0010011010010100
0111110110001111 0011100010001001 1111100001111001 0001111101010000
1001011111100000 0000111000011010 1100101110111000 0110111011110101
0110010111011010 1101010011010101 1010011011000111 0011010101001111
1000100100011101 0110011100011101 1101001010001100 0011110011110100
1010101000100100 1001010101000000 1010110001111110 0101111111111110
0001000001010101 0010000000011000 1101110100110110 0000011101100001
0110000101110000 1110001010110100 0010000111110000 0111101110111000

在传送之前，反向接入信道的反向 I 信道和反向 Q 信道都要被反向码序列直接序列扩展，这个反向码序列就是由长码序列和 W1 模 2 加运算得到的。

例 9.23　利用例 9.18 中 I 信道重复器输出结果，它与接入码模 2 加即可得到直接序列扩展。

I 信道重复器第一行输出为：

1111 1111 1111 0000 0000 1111 1111 1111

从例 9.22 中得到的接入码前两行是：

1110100000100001 0111101111100111 1100101110111110 0010011010010100
0111110110001111 0011100010001001 1111100001111001 0001111101010000

对应重复器输出的前两个符号 11，直接序列扩展的输出如下：

1→ 0001011111011110 1000010000011000 0011010001000001 1101100101101011

1→ 1000001001110000 1100011101110110 0000011110000110 1110000010101111

这个例子明确给出了计算直接序列扩展的方法。

例 9.24　利用例 9.19 中 Q 信道重复器输出结果，它与接入码模 2 加即可得到直接序列（DS）扩展。

Q 信道重复器第一行输出为：

1111 1111 1111 0000 0000 0000 1111 1111

比较 I 信道重复器输出结果与 Q 信道重复器输出结果，我们可以发现第 6 对数据不同，分别是 1111 和 0000。对应 Q 信道重复器输出的前两个符号 00，直接序列扩展的输出如下：

0→ 0110010001101111 1010010100101010 1101110100010011 1001010110011011

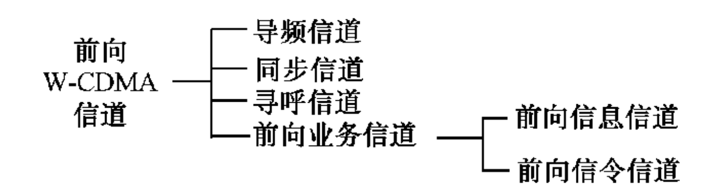
0→ 1101111100100111 1000101110001011 0101010110001001 0111101001000100

9.5.5 反向接入信道基带滤波

在节 9.3.5 中已经提到，经过直接序列扩展操作后，I，Q 接入信道脉冲要输出到 I，Q 信道基带滤波器。

第 10 章　前向 W-CDMA 信道

前向 W-CDMA 信道由以下码信道构成：(1)1 个导频信道；(2)1 个同步信道；(3)至多 8 个寻呼信道；(4)一定数量的前向业务信道。前向信道包括前向信息信道和前向信令信道。



每个码信道都要经过适当的 Walsh 函数和具有 4.096 Mc/s 固定码速率的 PN 序列正交扩展。

10.1　导频信道

导频信道被基站激活后就一直处于发送状态。导频信道是一种非调制的扩展信号，它用于移动台与所在覆盖区的基站取得同步，其结构如图 10.1 所示。

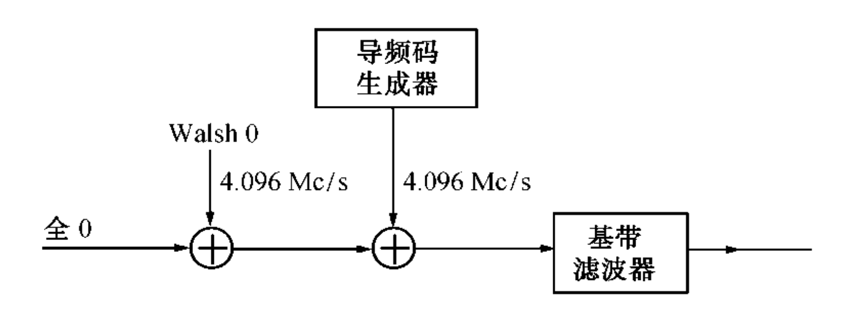


图 10.1　导频信道结构

每个基站用 1 个导频 PN 序列的时间偏移来识别前向信道，时间偏移可以在 PCS 系统中再利用。不同的导频信道通过从 0 到 319 的偏移编号来识别，这些偏移编号与零偏移导频 PN 序列相区别。

320 个惟一的值使导频 PN 序列偏移成为可能。从零偏移开始每个给定的导频 PN 序列在码片中的偏移量等于编号值乘 256，例如，导频 PN 序列的偏移编号为 15，则导频 PN 序列偏移是 15 × 256 = 3 480 个 PN 码片。对于一个给定的基站，所有频率指定都可以应用导频 PN 序列偏移。

10.1.1　导频信道正交和直接序列(DS)扩展

导频信道在发送前要与编号为 0 的 Walsh 码进行正交扩展。

为了对一个给定前向 W-CDMA 信道的所有码信道提供正交信道化，每个码信道都要用具有固定码速率 4.096 Mc/s 的 Walsh 码进行扩展。用编号为 n 的 Walsh 函数扩展的码信道指定为编号为 n 的码，编号为 0 的 Walsh 函数被指定为导频信道。

例 10.1　以 16 kb/s 数据速率输入的每帧 80 比特全零导频信道，码序列如下：

0000000000 0000000000 0000000000 0000000000

0000000000 0000000000 0000000000 0000000000

编号为 0 的 Walsh 函数由 64 个 0 组成：

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

由于每个 0 比特输入都要与 Walsh 码（64 个 0 码片）做扩展运算，因此所有与编号为 0 的 Walsh 函数正交扩展得到的符号共 80×64 = 5120 个（全为 0）。

例 10.2 PN 码生成器抽头多项式为：$g(x) = 1 + x + x^2 + x^{22} + x^{32}$，其向量形式为 111000000000000000000001000000001（见图 10.2），导频 PN 序列码速率为 4.096 Mc/s。长码存储元素设定为：10010011001010011011010111100011，将这两个向量输入图 10.2，生成的长码如下：

10111101 01110100 00101110 10110010 10011110 11101011 01110011 11000001

00101000 11011010 01101101 11011100 10101101 00101100 01001010 00000101

11000010 10110101 01011011 01001111 10011110 11101101 00111011 10100000

00110000 10001111 10000001 10000000 11110011 10010010 01100000 00011010

11011100 01001000 00110010 01001000 10000111 11011001 01101001 10100001

11111111 01110001 11000000 00010101 11111001 00101011 00001010 10101011

01000101 00000000 01110101 01001101 10001000 01100011 01010010 00110100

00110100 00100101 10110111 11100001 01110100 10100101 00101110 11101101

11110100 00000110 11001001 00110011 10000000 11110000 01000011 10001010

01100000 01001001 11011000

前面全零符号序列与编号为 0 的 Walsh 函数正交扩展的结果再与上面的长码序列进行模 2 加运算，就可得到直接序列扩展。然而，直接序列（DS）扩展的输出结果正是长码序列本身的重复。

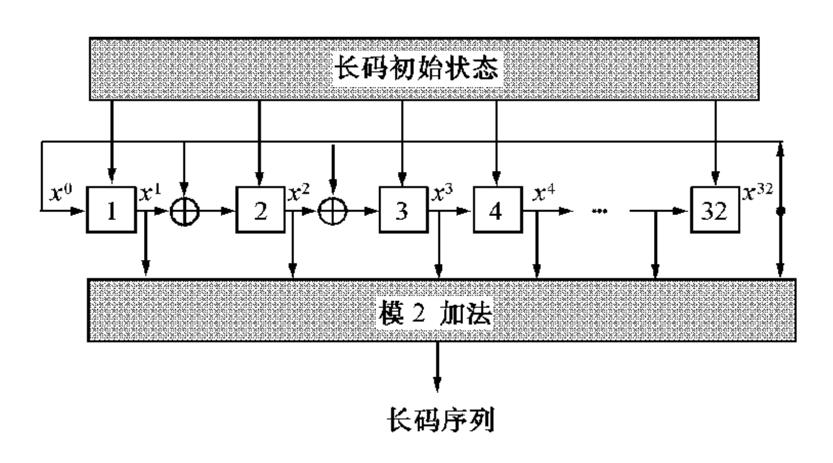


图 10.2 PN 码生成器生成长码序列

10.1.2 导频信道滤波

经过直接序列（DS）扩展操作后，导频信道还要经过如下滤波过程。如图 10.1 所示，直接序列扩展脉冲输出到基带滤波器。基带滤波器频率响应 $S(f)$ 满足图 10.3 所示的限制，滤波器的归一化频率响应的通带 $0 \leq f \leq f_p$，限制在 $\pm\delta_1$ 中，阻带 $f \geq f_s$，应小于或等于 $-\delta_2$，以上变量的值为 $\delta_1 = 1.5$ dB，$\delta_2 = 40$ dB，$f_p = 1.96$ MHz，$f_s = 2.47$ MHz。

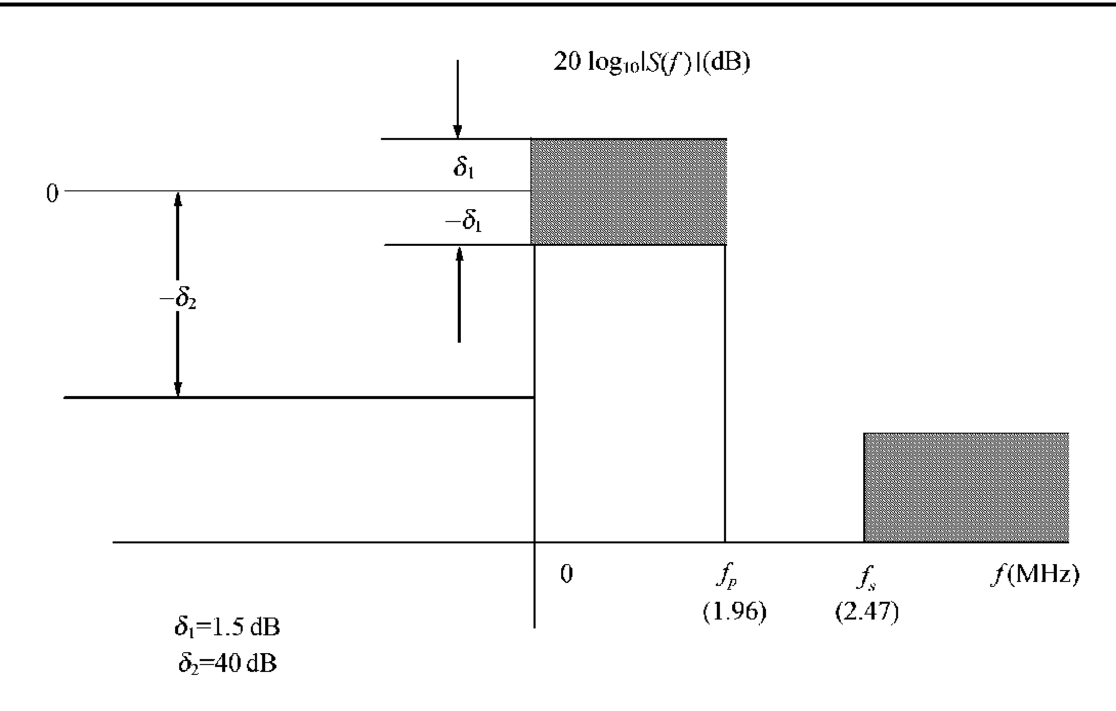


图 10.3　基带滤波器频率响应限制

10.2　同步信道

同步信道信号经过卷积编码、交织、扩展和调制后成为扩展信号，它用于在一个基站的覆盖区域中供移动台获取初始帧同步。

同步信道码速率为 16 kb/s，对一个给定的基站，同步信道的 I，Q 信道导频 PN 序列使用与导频信道相同的导频 PN 序列偏移。

由于同步信道由同一个导频序列扩展，并且同步信道的帧定时与导频 PN 序列相校准，所以一旦移动台通过获取导频信道得到导频 PN 序列，同步信道的同步信息就立刻可以得到。

应用于前向信道扩展的导频 PN 序列一启动，同步信道的块交织和帧就启动。

10.2.1　同步信道结构和调制参数

同步信道超帧的长度为 20 ms，同步信道信息只有在同步信道超帧起始的时候开始。同步信道整体结构如图 10.4 所示。

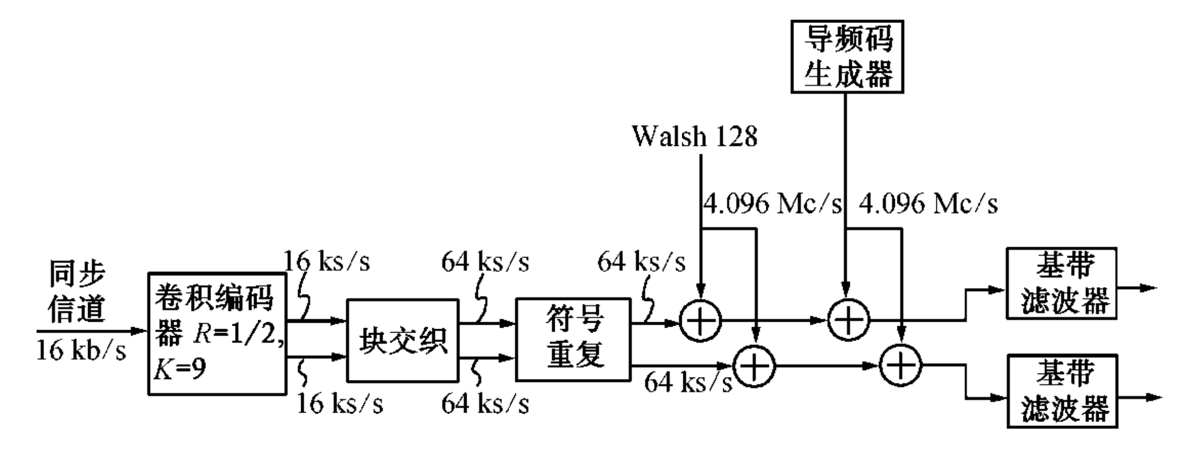


图 10.4　同步信道结构图[JTC(空中)/95 之后]

同步信道调制参数如下表 10.1。

表 10.1 同步信道调制参数

参　数	数据速率(b/s)16 000	单　位
PN 码速率	4.096	Mc/s
码率	1/2	比特/编码符号
符号重复	4	
符号速率	64 000	s/s
PN 码片/符号	64	PN 码片/符号
PN 码片/比特	256	PN 码片/比特

10.2.2 同步信道卷积编码

同步信道数据发送前要经过卷积编码,(2, 1, 8)卷积码具备码率 $R = 1/2$,约束长度 $K = 9$,生成抽头多项式分别如下:

$$g_0(x) = 1 + x + x^2 + x^3 + x^5 + x^7 + x^8$$

$$g_1(x) = 1 + x^2 + x^3 + x^4 + x^8$$

对应每个输入编码器的数据比特,(2, 1, 8)卷积码产生两个编码符号(c_0, c_1)。编码符号 c_0 先与生成多项式 $g_0(x)$编码并先被输出;之后,编码符号 c_1 与生成多项式 $g_1(x)$编码再被输出。卷积编码器在初始化前为全零状态。

卷积编码包括 1 个连续时间延迟序列选定的抽头的模 2 加。数据序列延迟长度为 $K-1 = 8$,图 10.5 给出了同步信道中用到的(2, 1, 8)卷积编码器。

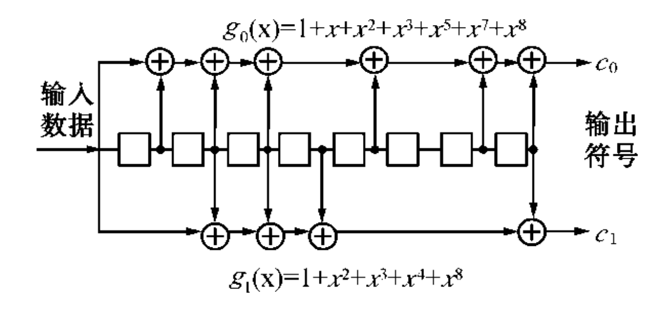


图 10.5 (2, 1, 8)卷积编码器

例 10.3 基站以16 kb/s的速率在同步信道中发送数据序列,每帧时长5 ms,包含 80 比特。

假设如下数据序列为编码器输入:

1001100101 0000110101 1000101101 1101001000

0110110010 1001110101 1001111000 0110011010

将数据如图 10.5 输入编码器,则卷积输出符号为:

c_0: 1110010111 1110111001 0110111110 1110110100

1101011110 0110010001 1011011100 0000101100

c_1: 1010010100 1101101001 0011010111 1110110000

1011010010 1111001011 0110011000 1000110111

10.2.3　同步信道交织

同步信道中编码后的符号 c_0 和 c_1 要经过如下交织过程：在调制和发送前基站要交织所有的码符号，一个块交织的时间跨度为5 ms。

同步信道的交织器是一个 8 行 10 列（80 个元素）的矩阵，表 10.2 和表 10.3 给出符号以 16 kb/s的速率分别写入 I，Q 同步信道的交织矩阵时写操作的顺序。同步信道编码符号从交织器中按行输出。

表 10.2　同步和寻呼 I 信道交织器输入（写入速率 16 kb/s，跨度 5 ms）

1	9	17	25	33	41	49	57	65	73
2	10	18	26	34	42	50	58	66	74
3	11	19	27	35	43	51	59	67	75
4	12	20	28	36	44	52	60	68	76
5	13	21	29	37	45	53	61	69	77
6	14	22	30	38	46	54	62	70	78
7	15	23	31	39	47	55	63	71	79
8	16	24	32	40	48	56	64	72	80

表 10.3　同步和寻呼 Q 信道交织器输入（写入速率 16 kb/s，跨度 5 ms）

37	45	53	61	69	77	5	13	21	29
38	46	54	62	70	78	6	14	22	30
39	47	55	63	71	79	7	15	23	31
40	48	56	64	72	80	8	16	24	32
41	49	57	65	73	1	9	17	25	33
42	50	58	66	74	2	10	18	26	34
43	51	59	67	75	3	11	19	27	35
44	52	60	68	76	4	12	20	28	36

例 10.4　依照表 10.2，按列将编码符号 c_0 写入 I 信道交织器：

1111111000
1101010010
1101100011
0111111110　⇨　**I 信道交织器输出（按行读）**
0101001101
1010110001
0111010100
1101011100

则 I 信道交织器输出结果为：

1111111000 1101010010 1101100011 0111111110
0101001101 1010110001 0111010100 1101011100

例 10.5　依照表 10.3，按列将编码符号 c_1 写入 Q 信道交织器：

0010000001
0111011101
0001110111
0000011011　⇨　**Q 信道交织器输出（按行读）**
1110010101
0001000010
1111111001
1110101111

则 Q 信道交织器输出结果为：

0010000001 0111011101 0001110111 0000011011

1110010101 0001000010 1111111001 1110101111

10.2.4 同步信道符号重复

在数据扰码之前，交织器以 16 kb/s 速率输出的符号重复 3 次(每个码元连续出现 4 次)。

例 10.6 参照例 10.4 中 I 信道交织器输出结果，每个符号重复 3 次，则符号重复结果如下：

1111 1111 1111 1111 1111 1111 1111 0000 0000 0000

1111 1111 0000 1111 0000 1111 0000 0000 1111 0000

1111 1111 0000 1111 1111 0000 0000 0000 1111 1111

0000 1111 1111 1111 1111 1111 1111 1111 1111 0000

0000 1111 0000 1111 0000 0000 1111 1111 0000 1111

1111 0000 1111 0000 1111 1111 0000 0000 0000 1111

例 10.7 Q 信道交织器输出结果如例 10.5，每个符号重复 3 次(每个码元连续出现 4 次)，则符号重复结果如下：

0000 0000 1111 0000 0000 0000 0000 0000 0000 1111

0000 1111 1111 1111 0000 1111 1111 1111 0000 1111

0000 0000 0000 1111 1111 1111 0000 1111 1111 1111

0000 0000 0000 0000 0000 1111 1111 0000 1111 1111

1111 1111 1111 0000 0000 1111 0000 1111 0000 1111

0000 0000 0000 1111 0000 0000 0000 0000 1111 0000

10.2.5 同步信道 Walsh 码扩展

前向 CDMA 信道的同步信道与具有固定码速率 4.096 Mc/s 的 Walsh128 扩展。同步信道中用来与 Walsh128 正交扩展的重复器输出符号的码速率为 64 kb/s，通过 Walsh128 与重复器输出符号在固定码速率 4.096 Mc/s 下的模 2 加来完成 Walsh 码扩展。

一般的，用编号为 n 的 Walsh 函数扩展的码信道被指定码编号 n($0\leqslant n\leqslant 64$ 的码速率为 64 kb/s，$0\leqslant n\leqslant 127$ 的码速率为 32 kb/s，$0\leqslant n\leqslant 255$ 的码速率为 16 kb/s)。表 10.4，10.5，10.6 中定义的时间正交 Walsh 码分别对应着 64 kb/s，32 kb/s，16 kb/s。在表 10.5，10.6 中的此处的文字为 $\overline{\text{Walsh64}}$，加上划线表示 Walsh64 的补码。

表 10.4 **64 kb/s 的 Walsh 码**

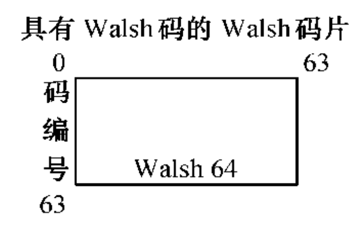


表 10.5 **32 kb/s 的 Walsh 码**

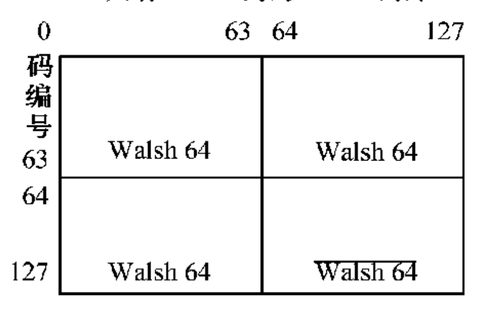
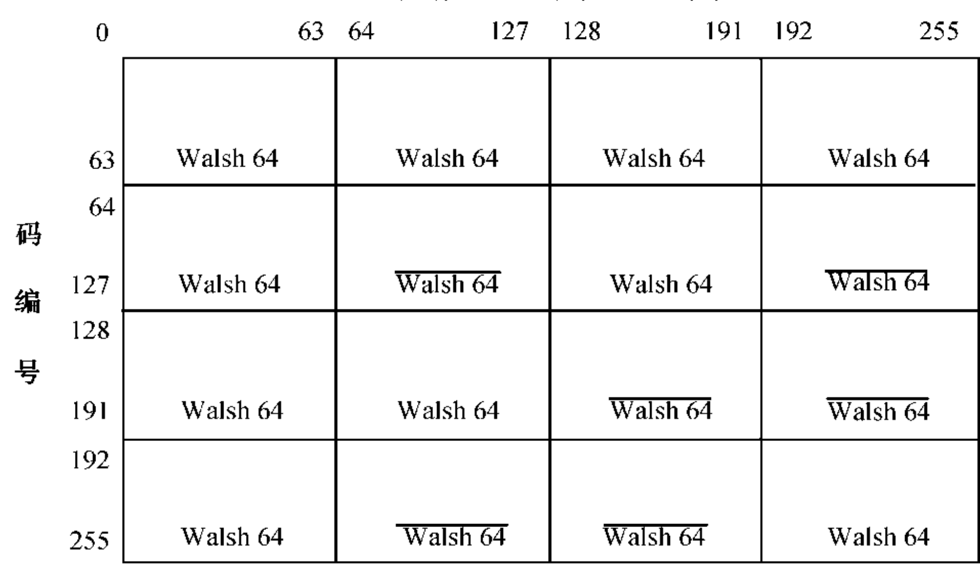


表 10.6 **16 kb/s 的 Walsh 码**



例 10.8 如下所示，Walsh128 被指定给 16 kb/s 编号为 $n=128$ 的码。

具有 Walsh码的 Walsh码片（0～255）

码编号 $128 \leqslant n \leqslant 191$				
128～191	Walsh 64	Walsh 64	$\overline{\text{Walsh 64}}$	$\overline{\text{Walsh 64}}$

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

例 10.9 如下所示，Walsh64 被指定给 32 kb/s 编号为 $n=64$ 的码。

Walsh码片（0～127）

0	127
Walsh 64	$\overline{\text{Walsh 64}}$

01100110 01100110 01100110 01100110

01100110 01100110 01100110 01100110

10011001 10011001 10011001 10011001

10011001 10011001 10011001 10011001

Walsh 64 的 Walsh 码片如表 10.7 所示，则$\overline{\text{Walsh64}}$的 Walsh 码片如表 10.8 所示。

表 10.7　**Walsh64 的 Walsh 码片**

具有 Walsh 码的 Walsh 码片

Walsh码编号	0000 0123	0000 4567	0011 8901	1111 2345	1111 6789	2222 0123	2222 4567	2233 8901	3333 2345	3333 6789	4444 0123	4444 4567	4455 8901	5555 2345	5555 6789	6666 0123
0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
1	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101
2	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011
3	0110	0110	0110	0110	0110	0110	0110	0110	0110	0110	0110	0110	0110	0110	0110	0110
4	0000	1111	0000	1111	0000	1111	0000	1111	0000	1111	0000	1111	0000	1111	0000	1111
5	0101	1010	0101	1010	0101	1010	0101	1010	0101	1010	0101	1010	0101	1010	0101	1010
6	0011	1100	0011	1100	0011	1100	0011	1100	0011	1100	0011	1100	0011	1100	0011	1100
7	0110	1001	0110	1001	0110	1001	0110	1001	0110	1001	0110	1001	0110	1001	0110	1001
8	0000	0000	1111	1111	0000	0000	1111	1111	0000	0000	1111	1111	0000	0000	1111	1111
9	0101	0101	1010	1010	0101	0101	1010	1010	0101	0101	1010	1010	0101	0101	1010	1010
10	0011	0011	1100	1100	0011	0011	1100	1100	0011	0011	1100	1100	0011	0011	1100	1100
11	0110	0110	1001	1001	0110	0110	1001	1001	0110	0110	1001	1001	0110	0110	1001	1001
12	0000	1111	1111	0000	0000	1111	1111	0000	0000	1111	1111	0000	0000	1111	1111	0000
13	0101	1010	1010	0101	0101	1010	1010	0101	0101	1010	1010	0101	0101	1010	1010	0101
14	0011	1100	1100	0011	0011	1100	1100	0011	0011	1100	1100	0011	0011	1100	1100	0011
15	0110	1001	1001	0110	0110	1001	1001	0110	0110	1001	1001	0110	0110	1001	1001	0110
16	0000	0000	0000	0000	1111	1111	1111	1111	0000	0000	0000	0000	1111	1111	1111	1111
17	0101	0101	0101	0101	1010	1010	1010	1010	0101	0101	0101	0101	1010	1010	1010	1010
18	0011	0011	0011	0011	1100	1100	1100	1100	0011	0011	0011	0011	1100	1100	1100	1100
19	0110	0110	0110	0110	1001	1001	1001	1001	0110	0110	0110	0110	1001	1001	1001	1001
20	0000	1111	0000	1111	1111	0000	1111	0000	0000	1111	0000	1111	1111	0000	1111	0000
21	0101	1010	0101	1010	1010	0101	1010	0101	0101	1010	0101	1010	1010	0101	1010	0101
22	0011	1100	0011	1100	1100	0011	1100	0011	0011	1100	0011	1100	1100	0011	1100	0011
23	0110	1001	0110	1001	1001	0110	1001	0110	0110	1001	0110	1001	1001	0110	1001	0110
24	0000	0000	1111	1111	1111	1111	0000	0000	0000	0000	1111	1111	1111	1111	0000	0000
25	0101	0101	1010	1010	1010	1010	0101	0101	0101	0101	1010	1010	1010	1010	0101	0101
26	0011	0011	1100	1100	1100	1100	0011	0011	0011	0011	1100	1100	1100	1100	0011	0011
27	0110	0110	1001	1001	1001	1001	0110	0110	0110	0110	1001	1001	1001	1001	0110	0110
28	0000	1111	1111	0000	1111	0000	0000	1111	0000	1111	1111	0000	1111	0000	0000	1111
29	0101	1010	1010	0101	1010	0101	0101	1010	0101	1010	1010	0101	1010	0101	0101	1010
30	0011	1100	1100	0011	1100	0011	0011	1100	0011	1100	1100	0011	1100	0011	0011	1100
31	0110	1001	1001	0110	1001	0110	0110	1001	0110	1001	1001	0110	1001	0110	0110	1001
32	0000	0000	0000	0000	0000	0000	0000	0000	1111	1111	1111	1111	1111	1111	1111	1111
33	0101	0101	0101	0101	0101	0101	0101	0101	1010	1010	1010	1010	1010	1010	1010	1010
34	0011	0011	0011	0011	0011	0011	0011	0011	1100	1100	1100	1100	1100	1100	1100	1100
35	0110	0110	0110	0110	0110	0110	0110	0110	1001	1001	1001	1001	1001	1001	1001	1001
36	0000	1111	0000	1111	0000	1111	0000	1111	1111	0000	1111	0000	1111	0000	1111	0000
37	0101	1010	0101	1010	0101	1010	0101	1010	1010	0101	1010	0101	1010	0101	1010	0101
38	0011	1100	0011	1100	0011	1100	0011	1100	1100	0011	1100	0011	1100	0011	1100	0011
39	0110	1001	0110	1001	0110	1001	0110	1001	1001	0110	1001	0110	1001	0110	1001	0110

（续表）

Walsh码编号																
40	0000	0000	1111	1111	0000	0000	1111	1111	1111	1111	0000	0000	1111	1111	0000	0000
41	0101	0101	1010	1010	0101	0101	1010	1010	1010	1010	0101	0101	1010	1010	0101	0101
42	0011	0011	1100	1100	0011	0011	1100	1100	1100	1100	0011	0011	1100	1100	0011	0011
43	0110	0110	1001	1001	0110	0110	1001	1001	1001	1001	0110	0110	1001	1001	0110	0110
44	0000	1111	1111	0000	0000	1111	1111	0000	1111	0000	0000	1111	1111	0000	0000	1111
45	0101	1010	1010	0101	0101	1010	1010	0101	1010	0101	0101	1010	1010	0101	0101	1010
46	0011	1100	1100	0011	0011	1100	1100	0011	1100	0011	0011	1100	1100	0011	0011	1100
47	0110	1001	1001	0110	0110	1001	1001	0110	1001	0110	0110	1001	1001	0110	0110	1001
48	0000	0000	0000	0000	1111	1111	1111	1111	1111	1111	1111	1111	0000	0000	0000	0000
49	0101	0101	0101	0101	1010	1010	1010	1010	1010	1010	1010	1010	0101	0101	0101	0101
50	0011	0011	0011	0011	1100	1100	1100	1100	1100	1100	1100	1100	0011	0011	0011	0011
51	0110	0110	0110	0110	1001	1001	1001	1001	1001	1001	1001	1001	0110	0110	0110	0110
52	0000	1111	0000	1111	1111	0000	1111	0000	1111	0000	1111	0000	0000	1111	0000	1111
53	0101	1010	0101	1010	1010	0101	1010	0101	1010	0101	1010	0101	0101	1010	0101	1010
54	0011	1100	0011	1100	1100	0011	1100	0011	1100	0011	1100	0011	0011	1100	0011	1100
55	0110	1001	0110	1001	1001	0110	1001	0110	1001	0110	1001	0110	0110	1001	0110	1001
56	0000	0000	1111	1111	1111	1111	0000	1111	1111	1111	0000	0000	0000	0000	1111	1111
57	0101	0101	1010	1010	1010	1010	0101	1010	1010	1010	0101	0101	0101	0101	1010	1010
58	0011	0011	1100	1100	1100	1100	0011	1100	1100	1100	0011	0011	0011	0011	1100	1100
59	0110	0110	1001	1001	1001	1001	0110	1001	1001	1001	0110	0110	0110	0110	1001	1001
60	0000	1111	1111	0000	1111	0000	0000	1111	1111	0000	0000	1111	0000	1111	1111	0000
61	0101	1010	1010	0101	1010	0101	0101	1010	1010	0101	0101	1010	0101	1010	1010	0101
62	0011	1100	1100	0011	1100	0011	0011	1100	1100	0011	0011	1100	0011	1100	1100	0011
63	0110	1001	1001	0110	1001	0110	0110	1001	1001	0110	0110	1001	0110	1001	1001	0110

表 10.8　**Walsh64 的 Walsh 码补码片**

具有 Walsh 码的 Walsh 码片

Walsh码编号	0000 0123	0000 4567	0011 8901	1111 2345	1111 6789	2222 0123	2222 4567	2233 8901	3333 2345	3333 6789	4444 0123	4444 4567	4455 8901	5555 2345	5555 6789	6666 0123
0	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111
1	1010	1010	1010	1010	1010	1010	1010	1010	1010	1010	1010	1010	1010	1010	1010	1010
2	1100	1100	1100	1100	1100	1100	1100	1100	1100	1100	1100	1100	1100	1100	1100	1100
3	1001	1001	1001	1001	1001	1001	1001	1001	1001	1001	1001	1001	1001	1001	1001	1001
4	1111	0000	1111	0000	1111	0000	1111	0000	1111	0000	1111	0000	1111	0000	1111	0000
5	1010	0101	1010	0101	1010	0101	1010	0101	1010	0101	1010	0101	1010	0101	1010	0101
6	1100	0011	1100	0011	1100	0011	1100	0011	1100	0011	1100	0011	1100	0011	1100	0011
7	1001	0110	1001	0110	1001	0110	1001	0110	1001	0110	1001	0110	1001	0110	1001	0110
8	1111	1111	0000	0000	1111	1111	0000	0000	1111	1111	0000	0000	1111	1111	0000	0000
9	1010	1010	0101	0101	1010	1010	0101	0101	1010	1010	0101	0101	1010	1010	0101	0101
10	1100	1100	0011	0011	1100	1100	0011	0011	1100	1100	0011	0011	1100	1100	0011	0011
11	1001	1001	0110	0110	1001	1001	0110	0110	1001	1001	0110	0110	1001	1001	0110	0110
12	1111	0000	0000	1111	1111	0000	0000	1111	1111	0000	0000	1111	1111	0000	0000	1111
13	1010	0101	0101	1010	1010	0101	0101	1010	1010	0101	0101	1010	1010	0101	0101	1010
14	1100	0011	0011	1100	1100	0011	0011	1100	1100	0011	0011	1100	1100	0011	0011	1100
15	1001	0110	0110	1001	1001	0110	0110	1001	1001	0110	0110	1001	1001	0110	0110	1001
16	1111	1111	1111	1111	0000	0000	0000	0000	1111	1111	1111	1111	0000	0000	0000	0000
17	1010	1010	1010	1010	0101	0101	0101	0101	1010	1010	1010	1010	0101	0101	0101	0101
18	1100	1100	1100	1100	0011	0011	0011	0011	1100	1100	1100	1100	0011	0011	0011	0011
19	1001	1001	1001	1001	0110	0110	0110	0110	1001	1001	1001	1001	0110	0110	0110	0110

（续表）

Walsh码编号																	
	20	1111	0000	1111	0000	0000	1111	0000	1111	1111	0000	1111	0000	0000	1111	0000	1111
	21	1010	0101	1010	0101	0101	1010	0101	1010	1010	0101	1010	0101	0101	1010	0101	1010
	22	1100	0011	1100	0011	0011	1100	0011	1100	1100	0011	1100	0011	0011	1100	0011	1100
	23	1001	0110	1001	0110	0110	1001	0110	1001	1001	0110	1001	0110	0110	1001	0110	1001
	24	1111	1111	0000	0000	0000	0000	1111	1111	1111	1111	0000	0000	0000	0000	1111	1111
	25	1010	1010	0101	0101	0101	0101	1010	1010	1010	1010	0101	0101	0101	0101	1010	1010
	26	1100	1100	0011	0011	0011	0011	1100	1100	1100	1100	0011	0011	0011	0011	1100	1100
	27	1001	1001	0110	0110	0110	0110	1001	1001	1001	1001	0110	0110	0110	0110	1001	1001
	28	1111	0000	0000	1111	0000	1111	1111	0000	1111	0000	0000	1111	0000	1111	1111	0000
	29	1010	0101	0101	1010	0101	1010	1010	0101	1010	0101	0101	1010	0101	1010	1010	0101
	30	1100	0011	0011	1100	0011	1100	1100	0011	1100	0011	0011	1100	0011	1100	1100	0011
	31	1001	0110	0110	1001	0110	1001	1001	0110	1001	0110	0110	1001	0110	1001	1001	0110
	32	1111	1111	1111	1111	1111	1111	1111	1111	0000	0000	0000	0000	0000	0000	0000	0000
	33	1010	1010	1010	1010	1010	1010	1010	1010	0101	0101	0101	0101	0101	0101	0101	0101
	34	1100	1100	1100	1100	1100	1100	1100	1100	0011	0011	0011	0011	0011	0011	0011	0011
	35	1001	1001	1001	1001	1001	1001	1001	1001	0110	0110	0110	0110	0110	0110	0110	0110
	36	1111	0000	1111	0000	1111	0000	1111	0000	0000	1111	0000	1111	0000	1111	0000	1111
	37	1010	0101	1010	0101	1010	0101	1010	0101	0101	1010	0101	1010	0101	1010	0101	1010
	38	1100	0011	1100	0011	1100	0011	1100	0011	0011	1100	0011	1100	0011	1100	0011	1100
	39	1001	0110	1001	0110	1001	0110	1001	0110	0110	1001	0110	1001	0110	1001	0110	1001
	40	1111	1111	0000	0000	1111	1111	0000	0000	0000	0000	1111	1111	0000	0000	1111	1111
	41	1010	1010	0101	0101	1010	1010	0101	0101	0101	0101	1010	1010	0101	0101	1010	1010
	42	1100	1100	0011	0011	1100	1100	0011	0011	0011	0011	1100	1100	0011	0011	1100	1100
	43	1001	1001	0110	0110	1001	1001	0110	0110	0110	0110	1001	1001	0110	0110	1001	1001
	44	1111	0000	0000	1111	1111	0000	0000	1111	0000	1111	1111	0000	0000	1111	1111	0000
	45	1010	0101	0101	1010	1010	0101	0101	1010	0101	1010	1010	0101	0101	1010	1010	0101
	46	1100	0011	0011	1100	1100	0011	0011	1100	0011	1100	1100	0011	0011	1100	1100	0011
	47	1001	0110	0110	1001	1001	0110	0110	1001	0110	1001	1001	0110	0110	1001	1001	0110
	48	1111	1111	1111	1111	0000	0000	0000	0000	0000	0000	0000	0000	1111	1111	1111	1111
	49	1010	1010	1010	1010	0101	0101	0101	0101	0101	0101	0101	0101	1010	1010	1010	1010
	50	1100	1100	1100	1100	0011	0011	0011	0011	0011	0011	0011	0011	1100	1100	1100	1100
	51	1001	1001	1001	1001	0110	0110	0110	0110	0110	0110	0110	0110	1001	1001	1001	1001
	52	1111	0000	1111	0000	0000	1111	0000	1111	0000	1111	0000	1111	1111	0000	1111	0000
	53	1010	0101	1010	0101	0101	1010	0101	1010	0101	1010	0101	1010	1010	0101	1010	0101
	54	1100	0011	1100	0011	0011	1100	0011	1100	0011	1100	0011	1100	1100	0011	1100	0011
	55	1001	0110	1001	0110	0110	1001	0110	1001	0110	1001	0110	1001	1001	0110	1001	0110
	56	1111	1111	0000	0000	0000	0000	1111	0000	0000	0000	1111	1111	1111	1111	0000	0000
	57	1010	1010	0101	0101	0101	0101	1010	0101	0101	0101	1010	1010	1010	1010	0101	0101
	58	1100	1100	0011	0011	0011	0011	1100	0011	0011	0011	1100	1100	1100	1100	0011	0011
	59	1001	1001	0110	0110	0110	0110	1001	0110	0110	0110	1001	1001	1001	1001	0110	0110
	60	1111	0000	0000	1111	0000	1111	1111	0000	0000	1111	1111	0000	1111	0000	0000	1111
	61	1010	0101	0101	1010	0101	1010	1010	0101	0101	1010	1010	0101	1010	0101	0101	1010
	62	1100	0011	0011	1100	0011	1100	1100	0011	0011	1100	1100	0011	1100	0011	0011	1100
	63	1001	0110	0110	1001	0110	1001	1001	0110	0110	1001	1001	0110	1001	0110	0110	1001

在下面的例子中，将讨论通过 Walsh n，$0 \leqslant n \leqslant 255$，与重复器输出符号在16 kb/s下的模 2 加来完成 Walsh 码扩展。

例 10.10　考虑利用 Walsh128 的扩展。

通过表 10.6，在例 10.8 中得到 Walsh128 码片，Walsh 码扩展计算如下：

$$扩展 = \text{I 信道重复符号} \oplus \text{Walsh128 码片}$$

由于例 10.6 中已给出 I 信道符号重复结果，例 10.8 已列出 Walsh128 码片，所以通过 Walsh128 的正交扩展如下：

符号重复　　　　Walsh 码扩展

1→ 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

1→ 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

1→ 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

1→ 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

这里只给出 4 个重复符号的 Walsh 码扩展。

例 10.11　考虑 Q 信道重复符号与 Walsh128 的扩展。

Q 信道重复符号在例 10.7 中已给出，因而 Q 信道重复符号与 Walsh128 模 2 加所得扩展结果如下：

符号重复　　　　Walsh 码扩展

0→ 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

0→ 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

0→ 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

0→ 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

这里只给出 Q 信道 4 个重复符号的 Walsh 码扩展。

10.2.6　同步信道正交相位扩展

Walsh 码扩展后，同步信道还要进行如图 10.4 所示的正交相位扩展。导频 PN 序列码速率为 4.096 Mc/s，1 个码序列经过 1 个周期 81 920 个码片的长码生成器生成了导频序列。具有$2^{32}-1$个码片的长码由抽头多项式 $g(x)=1+x+x^2+x^{22}+x^{32}$指定。长码生成器生成导频序列见图 10.2 所示。

例 10.12　生成器抽头向量为（11100000000000000000000000100000000001），长码存储元素向量为（10010011001010011011010111100011），如图 10.2 所示，用这两个向量可以得出导频码序列如下：

1011110101110100 0010111010110010 1001111011101011 0111001111000001
0010100011011010 0110110111011100 1010110100101100 0100101000000101
1100001010110101 0101101101001111 1001111011101101 0011101110100000
0011000010001111 1000000110000000 1111001110010010 0110000000011010
1101110001001000 0011001001001000 1000011111011001 0110100110100001
1111111111110001 1100000000010101 1111100100101011 0000101010101011
0100010100000000 0111010101001101 1000100001100011 0101001000110100
0011010000100101 1011011111100001 0111010010100101 0010111011101101
1111010000000110 1100100100110011 1000000011110000 0100001110001010
0110000001001001 11011000

得到导频码序列后，将其与 Walsh 码扩展进行模 2 加，便得到正交相位扩展。

例 10.13　利用同步 I 信道 Walsh 扩展与例 10.12 中得到的导频码进行模 2 加，可以得到正交相位扩展。计算形式如下：

正交相位扩展 = I 信道 Walsh 扩展 ⊕ 导频码序列

01000010 10001011 11010001 01001101 01100001 00010100 10001100 00111110
11010111 00100101 10010010 00100011 01010010 11010011 10110101 11111010
11000010 10110101 01011011 01001111 10011110 11101101 00111011 10100000
00110000 10001111 10000001 10000000 11110011 10010010 01100000 00011010
00100011 10110111 11001101 10110111 01111000 00100110 10010110 01011110
00000000 10001110 00111111 11101010 00000110 11010100 11110101 01010100
01000101 00000000 01110101 01001101 10001000 01100011 01010010 00110100
00110100 00100101 10110111 11100001 01110100 10100101 00101110 11101101

例 10.14　以同样方式，利用例 10.11 中得到的同步 Q 信道 Walsh 扩展与例 10.12 中得到的导频码进行模 2 加，可以得到正交相位扩展。计算形式如下：

正交相位扩展 = Q 信道 Walsh 扩展 ⊕ 导频码序列

10111101 01110100 00101110 10110010 10011110 11101011 01110011 11000001
00101000 11011010 01101101 11011100 10101101 00101100 01001010 00000101
00111101 01001010 10100100 10110000 01100001 00010010 11000100 01011111
11001111 01110000 01111110 01111111 00001100 01101101 10011111 11100101
11011100 01001000 00110010 01001000 10000111 11011001 01101001 10100001
11111111 01110001 11000000 00010101 11111001 00101011 00001010 10101011
10111010 11111111 10001010 10110010 01110111 10011100 10101101 11001011
11001011 11011010 01001000 00011110 10001011 01011010 11010001 00010010

10.2.7　同步信道基带滤波

在节 10.1.2 中提到，正交相位扩展之后，I, Q 脉冲要输入 I，Q 基带滤波器（见图 10.3），滤波后如图 10.6 所示将二进制数 I, Q 按照表 10.9 映射为不同相位。

表 10.9　前向 CDMA 信号的相位关系

I	Q	相位
0	0	$\frac{\pi}{4}$
1	0	$\frac{3\pi}{4}$
1	1	$-\frac{3\pi}{4}$
0	1	$-\frac{\pi}{4}$

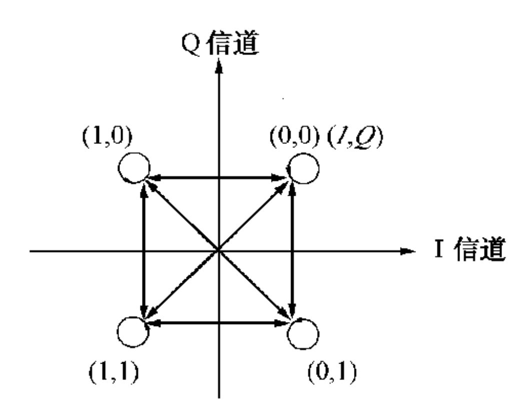


图 10.6　前向 CDMA 信道的信号相位点及其转换关系

10.3 寻呼信道

寻呼信道经过编码、交织、扩展和调制成为扩谱信号，移动台在基站覆盖区内监控寻呼信道。基站使用多个寻呼信道来传送系统信息和移动台特定信息。

寻呼信道以 16 kb/s 的固定数据速率发送信息，每帧时长 5 ms。寻呼信道的第一帧产生在基站传送开始，每个信道被分割成时长 20 ms 的信道时隙。

寻呼信道导频 PN 序列使用与给定的基站导频信道相同的导频 PN 序列偏移。图 10.7 给出了寻呼信道结构，表 10.10 给出了寻呼信道调制参数。

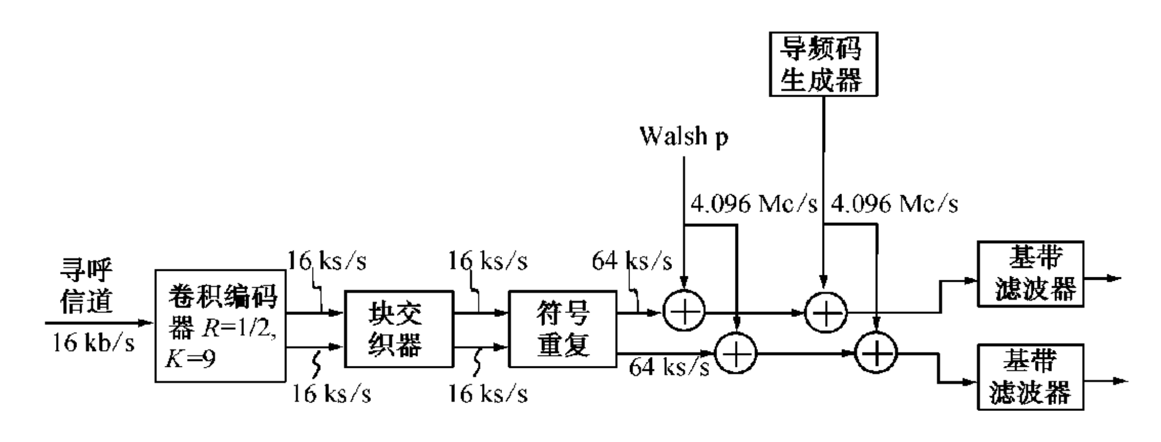


图 10.7 寻呼信道结构

表 10.10 寻呼信道调制参数

参 数	数据速率(b/s)16 000	单 位
PN 码速率	4.096	Mc/s
码率	1/2	比特/符号
符号重复	4	
符号速率	64 000	s/s
PN 码片/符号	64	PN 码片/符号
PN 码片/比特	256	PN 码片/比特

10.3.1 寻呼信道卷积编码

如节 10.2.2 中介绍，寻呼信道数据在发送之前要经过卷积编码。每比特输入(2，1，8)卷积编码器的数据都将产生两个编码符号(c_0，c_1)。

卷积编码器包括与下面两个抽头多项式相关的模 2 加运算：

$$g_0(x) = 1 + x + x^2 + x^3 + x^5 + x^7 + x^8$$

$$g_1(x) = 1 + x^2 + x^3 + x^4 + x^8$$

具有抽头多项式的(2，1，8)卷积编码器如图 10.5 所示。

例 10.15 基站在寻呼信道中以 16 kb/s 的速率发送数据，每帧时长 5 ms，包含 80 个比特。

假定输入编码器的序列如下：

1000110101 0001110101 1100101101 1101000001

0110110010 1001111101 1001011010 0110011010

依照每比特输入数据，可以计算出编码器的 c_0 和 c_1，(2，1，8)编码器输出如下：

c_0：1111110011 1000000011 1101010101 1110111010

0110010010 0110011110 1110000010 1011110100

c_1：1011011111 1001110101 0010100110 1110111010

0100110110 1111000000 1110010100 0100111111

10.3.2　寻呼信道交织

基站在调制和发送寻呼信道数据之前要对所有编码符号进行交织。寻呼信道中块交织的时间跨度为 5 ms。如表 10.2,10.3 所示，寻呼信道交织器是一个 8 行 10 列的矩阵，两表给出了 I,Q 信道中编码符号以 16 000 b/s 速率输入交织器时写操作的顺序。寻呼信道编码符号从交织器中按行输出。

例 10.16　依照表 10.2 填充 c_0 的所有编码符号，可以得到如下 8 行 10 列的矩阵：

1100101101
1101010101
1110110101
1011101001　⇨　**按行读取**
1010101110
1011010101
0001100110
0011000100

按行读出矩阵数据，则 I 信道交织器输出结果如下：

1100101101 1101010101 1110110101 1011101001

1010101110 1011010101 0001100110 0011000100

以上数据将输入 I 信道符号重复器。

例 10.17　依照表 10.3 应用例 10.15 中获得的 c_1，则通过将 c_1 写入 8×10 矩阵获得的 Q 信道块交织输出符号为：

1111010001
0111011100
1001011111
0100111101　⇨　**按行读取**
0100011011
1001001100
0100111001
0101110111

按行读取后，Q 信道的交织器输出为：

1111010001 0111011100 1001011111 0100111101

0100011011 1001001100 0100111001 0101110111

以上数据将输入 Q 信道符号重复器。

10.3.3 寻呼信道符号重复

寻呼信道的数据在经过卷积编码和交织后，在扰码之前要重复 3 次（每个码元连续出现 4 次）。

例 10.18 利用例 10.16 中获得的 I 信道交织器输出结果，每个符号重复 3 次的结果如下：

1111 1111 0000 0000 1111 0000 1111 1111 0000 1111
1111 1111 0000 1111 0000 1111 0000 1111 0000 1111
1111 1111 1111 0000 1111 1111 0000 1111 0000 1111
1111 0000 1111 1111 1111 0000 1111 0000 0000 1111
1111 0000 1111 0000 1111 0000 1111 1111 1111 0000
1111 0000 1111 1111 0000 1111 0000 1111 0000 1111
0000 0000 0000 1111 1111 0000 0000 1111 1111 0000
0000 0000 1111 1111 0000 0000 0000 1111 0000 0000

例 10.19 在例 10.17 中获得了 Q 信道交织器输出结果，则 Q 信道重复器对每个符号重复 3 次的结果如下：

1111 1111 1111 1111 0000 1111 0000 0000 0000 1111
0000 1111 1111 1111 0000 1111 1111 1111 0000 0000
1111 0000 0000 1111 0000 1111 1111 1111 1111 1111
0000 1111 0000 0000 1111 1111 1111 1111 0000 1111
0000 1111 0000 0000 0000 1111 1111 0000 1111 1111
1111 0000 0000 1111 0000 0000 1111 1111 0000 0000
0000 1111 0000 0000 1111 1111 1111 0000 0000 1111
0000 1111 0000 1111 1111 1111 0000 1111 1111 1111

10.3.4 寻呼信道正交扩展

寻呼信道发送之前要与具有同样编号的 Walsh 函数正交扩展。为了在一个给定的前向 W-CDMA 信道中为其所有信道提供正交信道，每个在前向 W-CDMA 信道中的码信道与具有固定码速率 4.096 Mc/s 的 Walsh 函数扩展。对于 16 kb/s 速率，与 Walsh n 扩展的寻呼信道被指定编号 n，$0 \leqslant n \leqslant 255$。在表 10.6 中定义了针对 16 kb/s 速率的 256 时间正交 Walsh 码。

每个基站有一个或几个（最多 7 个）寻呼信道，所以针对 16 kb/s 速率，至多有 256 个信道。如果 128 个 Walsh 码中的任意 1 个码被用在 32 kb/s 速率的信道上，则同一个码不能同时用在任何 16 kb/s 速率的信道上。

通过重复器输出和编号为 n（$1 \leqslant n \leqslant 7$）的 Walsh 码的模 2 加来完成正交扩展。

例 10.20 如果选择编号为 1 的 Walsh 码，则从表 10.6 可得具有 Walsh 码的 Walsh 码片如下：

01010101 01010101 01010101 01010101 01010101 01010101 01010101 01010101
01010101 01010101 01010101 01010101 01010101 01010101 01010101 01010101
01010101 01010101 01010101 01010101 01010101 01010101 01010101 01010101
01010101 01010101 01010101 01010101 01010101 01010101 01010101 01010101

I 信道重复器输出结果中第 8 至第 12 位的符号与编号为 1 的 Walsh 码模 2 加的结果如下：

符号编号　　　　　I 信道模 2 加结果

（8～12）

1→　10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101010

1→　10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101010

1→　10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101010

1→　10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101010

0→　01010101 01010101 01010101 01010101 01010101 01010101 01010101 01010101

0→　01010101 01010101 01010101 01010101 01010101 01010101 01010101 01010101

0→　01010101 01010101 01010101 01010101 01010101 01010101 01010101 01010101

0→　01010101 01010101 01010101 01010101 01010101 01010101 01010101 01010101

因而从上面的序列可以得到寻呼 I 信道的正交扩展结果。

例 10.21　从表 10.6 中选择编号为 1 的 Walsh 码，Q 信道重复器输出结果中第 8 至第 12 位的符号与其进行模 2 加的结果如下：

符号编号　　　　　Q 信道模 2 加结果

（8～12）

1→　10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101010

1→　10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101010

1→　10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101010

1→　10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101010

1→　10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101010

1→　10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101010

1→　10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101010

1→　10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101010

这些序列表示了基于寻呼 Q 信道的正交扩展。

10.3.5　寻呼信道正交相位调制

寻呼信道经过正交扩展最后还要进行正交相位扩展。导频 PN 序列码速率为4.096 Mc/s。从长码生成器生成的码序列经过一个周期的 81 920 个码片便生成一个导频序列（见图 10.2）。

例 10.22　依照图 10.2，导频码序列是由抽头多项式 $g(x)=1+x+x^2+x^{22}+x^{32}$ 定义的长码生成器生成的。

生成器抽头向量为：111000000000000000000001000000001

初始存储元素向量为：10010011001010011011010111100011

依照图 10.2 中的机制，这两个向量生成的导频码序列如下：

10111101 01110100 00101110 10110010 10011110 11101011 01110011 11000001
00101000 11011010 01101101 11011100 10101101 00101100 01001010 00000101
11000010 10110101 01011011 01001111 10011110 11101101 00111011 10100000
00110000 10001111 10000001 10000000 11110011 10010010 01100000 00011010
11011100 01001000 00110010 01001000 10000111 11011001 01101001 10100001
11111111 01110001 11000000 00010101 11111001 00101011 00001010 10101011
01000101 00000000 01110101 01001101 10001000 01100011 01010010 00110100
00110100 00100101 10110111 11100001 01110100 10100101 00101110 11101101
11110100 00000110 11001001 00110011 10000000 11110000 01000011 10001010
01100000 01001001 1101100

正交扩展后的符号序列与导频码序列的模 2 加即为正交相位扩展的结果。

例 10.23 利用在例 10.20 中得到的 I 信道正交调制序列和在例 10.22 中得到的导频码序列，可得到基于寻呼 I 信道的正交相位扩展序列如下：

00010111 11011110 10000100 00011000 00110100 01000001 11011001 01101011
10000010 01110000 11000111 01110110 00000111 10000110 11100000 10101111
01101000 00011111 11110001 11100101 00110100 01000111 10010001 00001010
10011010 00100101 00101011 00101010 01011001 00111000 11001010 10110000
10001001 00011101 01100111 00011101 11010010 10001100 00111100 11110100
10101010 00100100 10010101 01000000 10101100 01111110 01011111 11111110
00010000 01010101 00100000 00011000 11011101 00110110 00000111 01100001
01100001 01110000 11100010 10110100 00100001 11110000 01111011 10111000

例 10.24 利用在例 10.21 中得到的 Q 信道正交调制序列和在例 10.22 中得到的导频码序列，可得到基于寻呼 Q 信道的正交相位扩展序列如下：

00010111 11011110 10000100 00011000 00110100 01000001 11011001 01101011
10000010 01110000 11000111 01110110 00000111 10000110 11100000 10101111
01101000 00011111 11110001 11100101 00110100 01000111 10010001 00001010
10011010 00100101 00101011 00101010 01011001 00111000 11001010 10110000
01110110 11100010 10011000 11100010 00101101 01110011 11000011 00001011
01010101 11011011 01101010 10111111 01010011 10000001 10100000 00000001
11101111 10101010 11011111 11100111 00100010 11001001 11111000 10011110
10011110 10001111 00011101 01001011 11011110 00001111 10000100 01000111

10.3.6 寻呼信道滤波和映射

正交相位扩展操作之后，I,Q 脉冲输入如图 10.7 的 I,Q 基带滤波器。基带滤波器频率响应 $S(f)$满足图 10.3 所示的限制。

如图 10.6 所示，基带滤波后二进制的 I,Q 数据将按照表 10.9 映射为不同相位。

10.4 前向业务信道(**FTC**)

在一次呼叫过程中，前向业务信道将用户信息和信令信息发送给特定的移动台。前向信道同时支持的前向业务信道最大数目是：在 64 kb/s 的速率下为 64 个，32 kb/s 的速率下为 128 个，16 kb/s

的速率下为 256 个，在前向信道中的导频信道、寻呼信道和同步信道数目不包括在内。

基站以 64 kb/s，32 kb/s 和 16 kb/s 三种不同速率在前向信道中发送信息，截短多路器输出符号速率保持在 64 kb/s。前向业务信道每帧持续时间为 5 ms。对于给定的基站，前向业务信道和导频信道的导频 PN 序列具有相同的偏移。基站可以使用摆动前向业务信道帧。

前向业务信道的前向信息信道每帧发送速率为 64 kb/s，32 kb/s，16 kb/s，分别包含 320 比特，160 比特，80 比特，其偏移与反向业务信道时间偏移相同。前向信道的前向信息信道每帧发送速率为 4 kb/s 或 2 kb/s，分别包含 20 或 10 比特。

10.4.1　前向业务信道结构和调制参数

前向业务信道结构如图 10.8 所示，它是由一个前向信息信道和一个前向信令信道构成的。前向业务信道要与适当的 Walsh 码正交扩展，并与具有固定码速率为 4.096 Mc/s 的 PN 序列扩展。

前向业务信道的调制参数如表 10.11 和表 10.12：

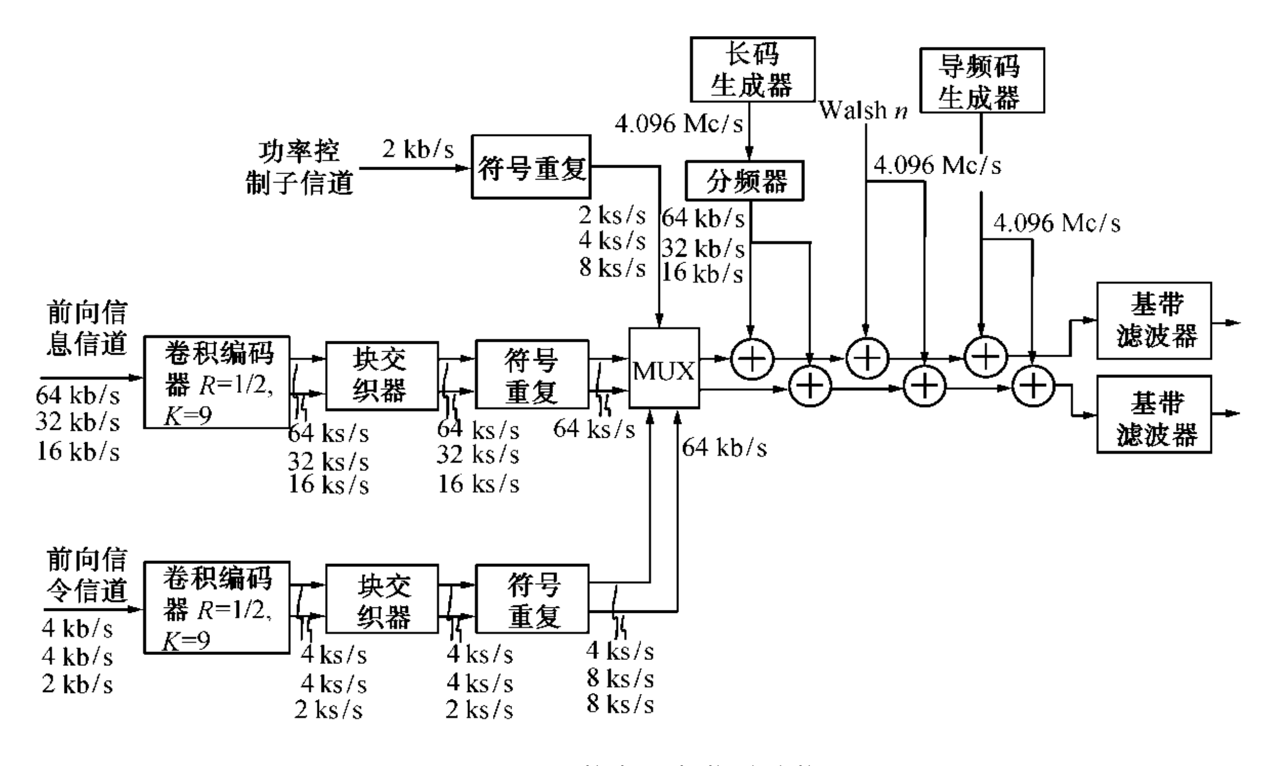


图 10.8　前向业务信道结构

表 10.11　前向业务信道的前向信息信道调制参数

参　数	数据速率(b/s)			单　位
	64 000	32 000	16 000	
PN 码速率	4.096	4.096	4.096	Mc/s
码率	1/2	1/2	1/2	比特/符号
符号重复	1	2	4	
截短比率	29/32	26/32	24/32	
有效码比率	16/29	8/13	4/6	比特/符号
符号速率	64 000	64 000	64 000	s/s
PN 码片/符号	64	64	64	PN 码片/符号
PN 码片/比特	58	104	192	PN 码片/比特

表 10.12　前向业务信道的前向信令信道调制参数

参　　数	数据速率(b/s)		单　　位
	4 000	**2 000**	
PN 码速率	4.096	4.096	Mc/s
码率	1/2	1/2	比特/符号
符号重复	1,2	4	
符号速率	4 000,8 000	8 000	s/s
PN 码片/符号	64	64	PN 码片/符号
PN 码片/比特	64,128	256	PN 码片/比特

10.4.2　前向业务信道卷积编码

在发送前,前向业务信道的信息信道和信令信道都要经过卷积编码。其中使用(2,1,8)卷积码,码率 $R = 1/2$,约束长度为 $K = 9$,生成器抽头多项式为:

$$g_0(x) = 1 + x + x^2 + x^3 + x^5 + x^7 + x^8$$

$$g_1(x) = 1 + x^2 + x^3 + x^4 + x^8$$

对于输入编码器的每个比特,(2,1,8)卷积编码器都将生成两个编码符号(c_0,c_1),通过 $g_0(x)$编码生成 c_0;通过 $g_1(x)$编码生成 c_1。卷积编码器的初始状态为全零。

卷积编码包括了一系列时间延迟序列选定抽头的模 2 加的过程,序列延迟长度为 $K - 1 = 8$。图 10.5 给出了前向业务信道中使用的(2,1,8)卷积编码器。

基站在前向业务信道中以 64 kb/s,32 kb/s 和 16 kb/s 三种不同速率发送信息,前向信令信道帧的发送速率为 4 kb/s 或 2 kb/s。

例 10.25　设基站以 16 kb/s 的数据速率在前向信息信道上发送数据,每个时长为 5 ms 的帧包括 80 比特。

假定输入编码器的序列如下:

1011000101 0110100011 1101010001 1001011100

1010011101 1101110001 1001011100 0100011011

(2,1,8)卷积编码器的初始状态为 100000000,其中的 1 是标识比特位。依照图 10.5 计算得出卷积编码符号如下:

c_0: 1101011111 0100111101 0100101100 0111010110

　　0100100100 1011011001 0001010110 1111101000

c_1: 1000000010 0000110000 1100010101 1000100011

　　1000110010 0101000011 1010100011 0100011110

10.4.3　FTC 交织

前向业务信道对信息信道和信令信道都设有交织器。在信息信道和信令信道上的 c_0,c_1 分别进行如下交织。在多路复用、调制和发送之前,基站要交织所有 FTC 信息信道和 FTC 信令信道上的编码符号。块交织器的时间跨度为 5/10/20 ms,具有 10 ms 或 20 ms 间隔的交织器为可选的,并只应用于 FTC 的信息信道。

FTC 信息信道交织器具有截短函数。FTC 信息信道速率为 64 kb/s，32 kb/s 或 16 kb/s 的数据输入分别对应的截短比率为 29/32，13/16 和 6/8，对应以上三种速率，交织器分别为 29 行 10 列、10 行 13 列及 6 行 10 列的矩阵。

FTC 信令信道的交织器对应 4 kb/s 速率为 5 行 4 列（20 个元素）的矩阵，对应 2 kb/s 为 5 行 2 列（10 个元素）的矩阵。

在表 10.13 和 10.14 中给出符号以 16 kb/s 的速率输入 I，Q 信息信道交织矩阵时写操作的顺序。前向业务信道编码符号从交织器中按行输出。

表 10.13　**FTC** 信息 **I** 信道交织器输入

（写操作速率为 **16 000 b/s**；交织器跨度为 **5 ms**）

1	9	17	25	33	41	49	57	65	73
2	10	18	26	34	42	50	58	66	74
3	11	19	27	35	43	51	59	67	75
5	13	21	29	37	45	53	61	69	77
6	14	22	30	38	46	54	62	70	78
7	15	23	31	39	47	55	63	71	79

表 10.14　**FTC** 信息 **Q** 信道交织器输入

（写操作速率为 **16 000 b/s**；交织器跨度为 **5 ms**）

37	45	53	61	69	77	5	13	21	29
39	47	55	63	71	79	7	15	23	31
40	48	56	64	72	80	8	16	24	32
41	49	57	65	73	1	9	17	25	33
43	51	59	67	75	2	11	19	27	35
44	52	60	68	76	4	12	20	28	36

注意：对于 16 kb/s 速率，前向业务信道的信息 I，Q 信道截短比率为 6/8，即 6 行 10 列，（从 80 个元素到 60 个元素）。

例 10.26　利用例 10.25 中得到的编码符号序列 c_0，依照表 10.13 将其按列输入信息 I 信道交织器。

I 信道交织器按列输入

⇩

1111100101
1110110011
0001001001　　⇨　**交织器按行输出**
0000011011
1010101000
1100100010

则信息 I 信道交织器输出序列为：

1111100101 1110110011 0001001001

0000011011 1010101000 1100100010

例 10.27　依照表 10.14 将编码符号序列 c_1 按列输入信息 Q 信道交织器。

则信息 Q 信道交织器输出序列为：

Q信道交织器按列输入

⇩

0101110010
1011010101
1000100100
1101011000
0000000001
0110100010

➩ 交织器按行输出

0101110010 1011010101 1000100100
1101011000 0000000001 0110100010

10.4.4 FTC 符号重复

交织后，前向业务信道数据根据不同速率进行重复。在多路复用前 64 kb/s 的数据不能重复；32 kb/s 的符号重复 1 次（每个码元连续出现 2 次）；16 kb/s 的符号重复 3 次（每个码元连续出现 4 次）。

例 10.28 每个速率为 16 kb/s 的符号要重复 3 次，利用例 10.26 中得到的信息 I 信道交织器符号输出，则 FTC 信息 I 信道符号重复为：

1111 1111 1111 1111 1111 0000 0000 1111 0000 1111
1111 1111 1111 0000 1111 1111 0000 0000 1111 1111
0000 0000 0000 1111 0000 0000 1111 0000 0000 1111
0000 0000 0000 0000 0000 1111 1111 0000 1111 1111
1111 0000 1111 0000 1111 0000 1111 0000 0000 0000
1111 1111 0000 0000 1111 0000 0000 0000 1111 0000

例 10.29 利用例 10.27 中得到的信息 Q 信道交织器符号输出，则 FTC 信息 Q 信道符号重复为：

0000 1111 0000 1111 1111 1111 0000 0000 1111 0000
1111 0000 1111 1111 0000 1111 0000 1111 0000 1111
1111 0000 0000 0000 1111 0000 0000 1111 0000 0000
1111 1111 0000 1111 0000 1111 1111 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 1111
0000 1111 1111 0000 1111 0000 0000 0000 1111 0000

10.4.5 前向信令信道卷积编码

在发送前，前向业务信道（FTC）的信令信道要经过卷积编码。其中使用（2，1，8）卷积码的生成器抽头多项式为：

$$g_0(x) = 1 + x + x^2 + x^3 + x^5 + x^7 + x^8$$

$$g_1(x) = 1 + x^2 + x^3 + x^4 + x^8$$

依照图 10.5，对于输入编码器的每个比特，（2，1，8）卷积编码器都将生成编码符号对（c_0，c_1）。前向业务信道的信令 I，Q 信道帧数据速率为 4 kb/s 或 2 kb/s。

例 10.30　数据速率为 2 kb/s 的前向信息 I,Q 信道每帧包含 10 比特。设输入(2，1，8)卷积编码器的数据为 1001110001,寄存器初始值为全零。则输出的编码符号如下：

偏移编号	寄存器内容	编码符号
0	100000000	(1,1)
1	010000000	(1,0)
2	001000000	(1,1)
3	100100000	(0,0)
4	110010000	(0,0)
5	111001000	(0,0)
6	011100100	(1,0)
7	001110010	(1,1)
8	000111001	(1,1)
9	100011100	(0,0)

编码符号 c_0 的生成多项式为：

$$g_0(x)=1+x+x^2+x^3+x^5+x^7+x^8$$

$$c_0=(1110001110)$$

编码符号 c_1 的生成多项式为：

$$g_1(x)=1+x^2+x^3+x^4+x^8$$

$$c_1=(1010000110)$$

10.4.6　前向信令信道交织

前向业务信道具有针对信令的交织器,块交织时间跨度为 5 ms。如表 10.15,FTC 的前向信令信道交织器是一个速率为 2 kb/s、5 行 2 列(10 个元素)的矩阵。

表 10.15　前向信令信道交织器输入
(在 **2 000 b/s** 速率下 **I,Q** 信道的写操作)

1	6
2	7
3	8
4	9
5	10

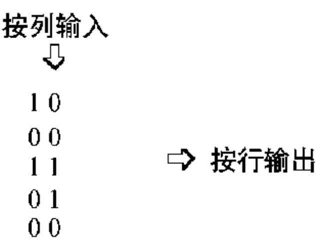
例 10.31　利用例 10.30 中得到的编码符号 $c_0=(1110001110)$,将其按列写入如表 10.15 的前向信令 I 信道交织器：

按列输入
⇩
1 0
1 1
1 1　　⇨　**按行输出**
0 1
0 0

则前向信令 I 信道交织器输出为:1011110100

例 10.32　利用例 10.30 中得到的编码符号 $c_1=(1010000110)$,将其按列写入如表 10.15 的前向

信令 Q 信道交织器：



则前向信令 Q 信道交织器输出为：1000110100

10.4.7 前向信令信道符号重复

在交织后、多路复用之前，前向信令信道符号根据不同速率进行交织，4 kb/s 的符号重复 1 次（每个码元连续出现 2 次）。

例 10.33 速率为 2 kb/s 的符号重复 3 次。从前向信令 I，Q 信道交织器输出的符号重复如下：

1. FTC 信令 I 信道符号重复：

利用例 10.31 得到的 I 信令信道交织器输出，则 I 信道符号重复如下：

1111 0000 1111 1111 1111 1111 0000 1111 0000 0000

2. FTC 信令 Q 信道符号重复：

利用例 10.32 得到的 Q 信令信道交织器输出，则 Q 信道符号重复如下：

1111 0000 0000 0000 1111 1111 0000 1111 0000 0000

10.4.8 FTC 功率控制子状态

基站产生功率控制比特，并将其插入到每个前向业务信道中。功率控制子信道在前向业务信道中不间断地发送，速率为每 500 μs 发送 1 个比特（即 2 kb/s）。

平均功率参照指定的具有 5.0 MHz 带宽的 CDMA 信道。功率在基站平均输入功率和平均输出功率之间取值，将之定义为一个常量以便于控制移动台的平均输出功率。

0 比特表示移动台增加输出功率；1 表示移动台减少平均输出功率。移动台针对每个功率控制比特增加或减少功率的数量是特定的。

基站的反向业务信道接收器估算接受到的特定移动台的信号强度，移动台指定周期为500 μs，恰好等于 32 个编码符号。基站接收器利用得到的估算结果决定功率控制比特的值为 0 或 1。基站相应地在如图 10.8 所示利用功率控制子信道的前向业务信道中发送功率控制比特。

10.4.9 前向业务信道多路复用

前向业务信道多路复用的有效符号输出速率为 64 ks/s。FTC 的前向信息信道、前向信令信道、功率控制子信道的多路复用过程如下。符号重复后对应 64 kb/s，32 kb/s 和 16 kb/s 三种不同速率，基站依照表 10.16，10.17，10.18 对功率控制比特、信令比特和信息比特进行多路复用。在这几个表中，p，s，i 分别代表功率控制比特、信令符号和信息符号。MUX 叫做复用的前向业务信道。

表 10.16 前向业务信道多路复用矩阵

（在 64 kb/s 下的写操作）

p1	s1	s2	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	i11	i12	i13

i14	i15	i16	i17	i18	i19	i20	i21	i22	i23	i24	i25	i26	i27	i28	i29
p2	s3	s4	i30	i31	i32	i33	i34	i35	i36	i37	i38	i39	i40	i41	i42
i43	i44	i45	i46	i47	i48	i49	i50	i51	i52	i53	i54	i55	i56	i57	i58
p3	s5	s6	i59	i60	i61	i62	i63	i64	i65	i66	i67	i68	i69	i70	i71
i72	i73	i74	i75	i76	i77	i78	i79	i80	i81	i82	i83	i84	i85	i86	i87
p4	s7	s8	i88	i89	i90	i91	i92	i93	i94	i95	i96	i97	i98	i99	i100
i101	i102	i103	i104	i105	i106	i107	i108	i109	i110	i111	i112	i113	i114	i115	i116
p5	s5	s10	i117	i118	i119	i120	i121	i122	i123	i124	i125	i126	i127	i128	i129
i130	i131	i132	i133	i134	i135	i136	i137	i138	i139	i140	i141	i142	i143	i144	i145
p6	s11	s12	i146	i147	i148	i149	i150	i151	i152	i153	i154	i155	i156	i157	i158
i159	i160	i161	i162	i163	i164	i165	i166	i167	i168	i169	i170	i171	i172	i173	i174
p7	s13	s14	i175	i176	i177	i178	i179	i180	i181	i182	i183	i184	i185	i186	i187
i188	i189	i190	i191	i192	i193	i194	i195	i196	i197	i198	i199	i200	i201	i202	i203
p8	s15	s16	i204	i205	i206	i207	i208	i209	i210	i211	i212	i213	i214	i215	i216
i217	i218	i219	i220	i221	i222	i223	i224	i225	i226	i227	i228	i229	i230	i231	i232
p9	s17	s18	i233	i234	i235	i236	i237	i238	i239	i240	i241	i242	i243	i244	i245
i246	i247	i248	i249	i250	i251	i252	i253	i254	i255	i256	i257	i258	i259	i260	i261
p10	s19	s20	i262	i263	i264	i265	i266	i267	i268	i269	i270	i271	i272	i273	i274
i275	i276	i277	i278	i279	i280	i281	i282	i283	i284	i285	i286	i287	i288	i289	i290

表 10.17　前向业务信道多路复用矩阵

（在 32 kb/s 速率下的写操作）

p1	p1	s1	s1	s2	s2	i1	i1	i2	i2	i3	i3	i4	i4	i5	i5
i6	i6	i7	i7	i8	i8	i9	i9	i10	i10	i11	i11	i12	i12	i13	i13
p2	p2	s3	s3	s4	s4	i14	i14	i15	i15	i16	i16	i17	i17	i18	i18
i19	i19	i20	i20	i21	i21	i22	i22	i23	i23	i24	i24	i25	i25	i26	i26
p3	p3	s5	s5	s6	s6	i27	i27	i28	i28	i29	i29	i30	i30	i31	i31
i32	i32	i33	i33	i34	i34	i35	i35	i36	i36	i37	i37	i38	i38	i39	i39
p4	p4	s7	s7	s8	s8	i40	i40	i41	i41	i42	i42	i43	i43	i44	i44
i45	i45	i46	i46	i47	i47	i48	i48	i49	i49	i50	i50	i51	i51	i52	i52
p5	p5	s9	s9	s10	s10	i53	i53	i54	i54	i55	i55	i56	i56	i57	i57
i58	i58	i59	i59	i60	i60	i61	i61	i62	i62	i63	i63	i64	i64	i65	i65
p6	p6	s11	s11	s12	s12	i66	i66	i67	i67	i68	i68	i69	i69	i70	i70
i71	i71	i72	i72	i73	i73	i74	i74	i75	i75	i76	i76	i77	i77	i78	i78
p7	p7	s13	s13	s14	s14	i79	i79	i80	i80	i81	i81	i82	i82	i83	i83
i84	i84	i85	i85	i86	i86	i87	i87	i88	i88	i89	i89	i90	i90	i91	i91
p8	p8	s15	s15	s16	s16	i92	i92	i93	i93	i94	i94	i95	i95	i96	i96
i97	i97	i98	i98	i99	i99	i100	i100	i101	i101	i102	i102	i103	i103	i104	i104
p9	p9	s17	s17	s18	s18	i105	i105	i106	i106	i107	i107	i108	i108	i109	i109
i110	i110	i111	i111	i112	i112	i113	i113	i114	i114	i115	i115	i116	i116	i117	i117
p10	p10	s19	s19	s20	s20	i118	i118	i119	i119	i120	i120	i121	i121	i122	i122
i123	i123	i124	i124	i125	i125	i126	i126	i127	i127	i128	i128	i129	i129	i130	i130

表 10.18　前向业务信道多路复用矩阵

（在 16 kb/s 速率下的写操作）

p1	p1	p1	p1	s1	s1	s1	s1	i1	i1	i1	i1	i2	i2	i2	i2
i3	i3	i3	i3	i4	i4	i4	i4	i5	i5	i5	i5	i6	i6	i6	i6
p2	p2	p2	p2	s2	s2	s2	s2	i7	i7	i7	i7	i8	i8	i8	i8
i9	i9	i9	i9	i10	i10	i10	i10	i11	i11	i11	i11	i12	i12	i12	i12
p3	p3	p3	p3	s3	s3	s3	s3	i13	i13	i13	i13	i14	i14	i14	i14
i15	i15	i15	i15	i16	i16	i16	i16	i17	i17	i17	i17	i18	i18	i18	i18
p4	p4	p4	p4	s4	s4	s4	s4	i19	i19	i19	i19	i20	i20	i20	i20
i21	i21	i21	i21	i22	i22	i22	i22	i23	i23	i23	i23	i24	i24	i24	i24
p5	p5	p5	p5	s5	s5	s5	s5	i25	i25	i25	i25	i26	i26	i26	i26
i27	i27	i27	i27	i28	i28	i28	i28	i29	i29	i29	i29	i30	i30	i30	i30
p6	p6	p6	p6	s6	s6	s6	s6	i31	i31	i31	i31	i32	i32	i32	i32
i33	i33	i33	i33	i34	i34	i34	i34	i35	i35	i35	i35	i36	i36	i36	i36
p7	p7	p7	p7	s7	s7	s7	s7	i37	i37	i37	i37	i38	i38	i38	i38
i39	i39	i39	i39	i40	i40	i40	i40	i41	i41	i41	i41	i42	i42	i42	i42
p8	p8	p8	p8	s8	s8	s8	s8	i43	i43	i43	i43	i44	i44	i44	i44
i45	i45	i45	i45	i46	i46	i46	i46	i47	i47	i47	i47	i48	i48	i48	i48
p9	p9	p9	p9	s9	s9	s9	s9	i49	i49	i49	i49	i50	i50	i50	i50
i51	i51	i51	i51	i52	i52	i52	i52	i53	i53	i53	i53	i54	i54	i54	i54
p10	p10	p10	p10	s10	s10	s10	s10	i55	i55	i55	i55	i56	i56	i56	i56
i57	i57	i57	i57	i58	i58	i58	i58	i59	i59	i59	i59	i60	i60	i60	i60

p：功率控制比特，s：信令符号，i：信息符号[JTC(AIR)/95.02.02－037R1 之后]

例 10.34　对于码率为 16 kb/s 的前向业务信道，在符号交织后，功率控制比特、信令比特和信息比特的多路复用如表 10.18 所示。

- 假定功率控制比特为：1000010011，即对应 p1＝1，p2＝0，p3＝0，p4＝0，p5＝0，p6＝1，p7＝0，p8＝0，p9＝1，p10＝1。
- 前向信令 I 信道交织器输出为：1011110100，即对应 s1＝1，s2＝0，s3＝1，s4＝1，s5＝1，s6＝1，s7＝0，s8＝1，s9＝0，s10＝0。
- 前向信息 I 信道交织器输出为：

  i1　i5　　i10　　i15　　i20　　i25　　i30

  1111100101　1110110011　0001001001

  　i35　　i40　　i45　　i50　　i55　　i60

  0000011011　1010101000　1100100010

将这些功率控制比特、I 信道信令比特和 I 信道信息比特替换到表 10.18 的前向 I 或 Q 业务信道多路复用矩阵，生成的多路复用前向业务 I 信道如下：

I 信道多路复用符号

1111 1111 1111 1111

1111 1111 1111 0000

0000 0000 0000 1111

0000 1111 1111 1111

0000 1111 1111 0000

1111 1111 0000 0000

0000 1111 1111 1111

0000 0000 0000 1111

0000 1111 0000 0000

1111 0000 0000 1111

1111 1111 0000 0000

0000 0000 0000 1111

0000 0000 1111 0000

1111 1111 1111 0000

0000 1111 1111 0000

1111 0000 1111 0000

1111 0000 0000 0000

1111 1111 0000 0000

1111 0000 1111 0000

0000 0000 1111 0000

以上为前向业务 I 信道的多路复用符号。

例 10.35　在符号交织后，功率控制比特、Q 信道信令比特和 Q 信道信息比特如表 10.18 的多路复用，生成前向业务 Q 信道多路复用符号序列。

- 功率控制比特：

  1000010011 ↔ p1，p2，p3，p4，p5，p6，p7，p8，p9，p10
- 前向信令 Q 信道交织器输出为：

  1000110100 ↔ s1，s2，s3，s4，s5，s6，s7，s8，s9，s10
- 前向信息 Q 信道交织器输出为：

  i1　i5　　i10　i15　i20　　i25　i30

  0101110010　1011010101　1000100100

  　i35　i40　　i45　i50　　i55　i60

  1101011000　0000000001　0110100010

将这些功率控制比特、Q 信道信令比特和 Q 信道信息比特替换到表 10.18 的多路复用矩阵，结果如下：

Q 信道多路复用符号

1111 1111 0000 1111

0000 1111 1111 1111

0000 0000 0000 0000

1111 0000 1111 0000

0000 0000 1111 1111

0000 1111 0000 1111

0000 0000 0000 1111

1111 0000 0000 0000

0000 1111 1111 0000

0000 1111 0000 0000

1111 1111 1111 1111

0000 1111 0000 1111

0000 0000 1111 0000

0000 0000 0000 0000

0000 1111 0000 0000

0000 0000 0000 0000

1111 0000 0000 1111

0000 1111 1111 0000

1111 0000 1111 0000

0000 0000 1111 0000

10.4.10 不连续发送

当可以实现不连续发送功能时，在发送之前，前向业务信道要经过一个时间选通器，这个选通器可以发送确定的截短、多路复用的输出符号，并根据语音激活指示器删除其他符号。语音激活探测器（VAD）用来为一个有效的信号确认一个可用的业务信道。语音激活探测器的目的是为了拒绝输入信道的高阶噪声，避免引入信号前端去噪，使脉冲噪音的误操作最小化，编码信令序列的去噪等等。信令与功率控制比特一直处于发送状态。如图 10.9 所示：信令和功率控制比特循环对于 64 kb/s，32 kb/s，16 kb/s 的发送速率分别选通（即发送）46.81 μs，93.75 μs，125 μs，非选通（即不发送）453.125 μs，406.25 μs，375 μs。循环提供每帧 10 次重复。

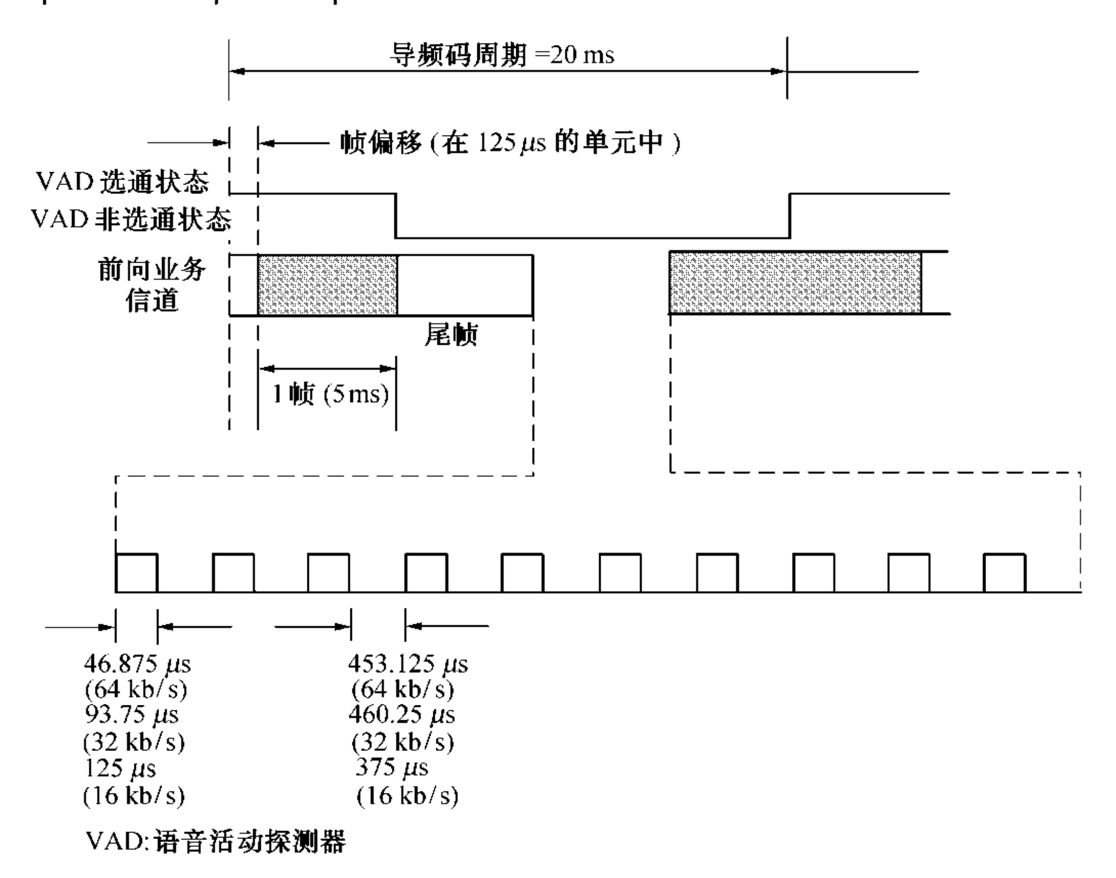


图 10.9　非连续发送示例

每个前向业务信道具有 1 024 个码片（125 μs）的 0，1，2 或 3 个单元的偏移，以移位每个前向信道不连续发送的选通时间。基站为每个前向信道选定合适的帧偏移。在一个呼叫过程中，基站

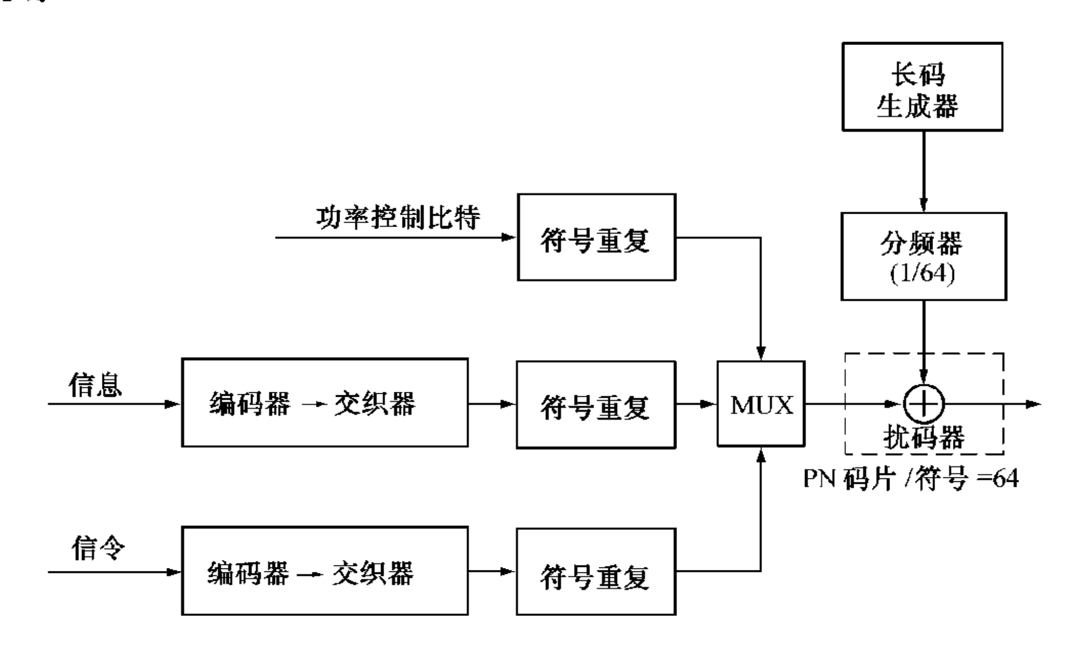
将帧偏移信息发送给每个移动台。

当一帧通过语音激活探测器被检测为非选通(即不发送)状态，下一帧的输入数据即被编码为 0。它被称为尾帧，为卷积编码解码器提供尾比特。

当移动台检测到语音激活探测器的发送状态由非选通(OFF)转为选通(ON)时，卷积编码器将起始帧设为全零状态。

10.4.11　前向业务信道数据扰码

前向业务信道中应用了扰码技术，FTC 在多路复用后进行扰码，即在每个多路复用器输出符号调制的基础上进行扰码。如图 10.10(速率为 64 ks/s)所示，通过多路复用后的输出符号与长码 PN 序列二进制值(0 或 1)的模 2 加便实现了扰码过程。对于 64 kb/s，32 kb/s，16 ks/s 的速率进行抽取所得长码 PN 序列的输出速率分别为 64 kb/s，32 kb/s，16 kb/s。这个 PN 序列等于在 4.096 MHz 时钟速率下的长码操作，在这一操作中只有每 64 个码片的第一个输出用在数据扰码中。长码的生成过程如下说明。



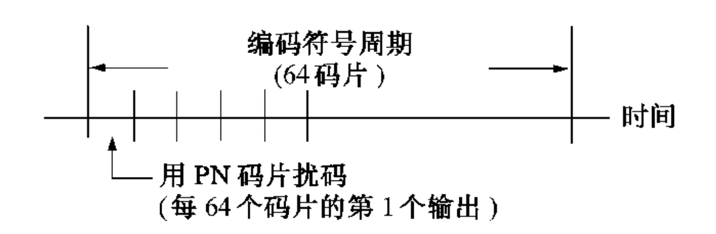


图 10.10　数据扰码函数

例 10.36　长码存储元素在基站的寻呼信道中发送，假定 32 位的初始存储元素向量为：10010011001010011011010111100011。

依照图 10.2，PN 码生成器的寄存器抽头多项式为 $g(x)=1+x+x^2+x^{22}+x^{32}$，其向量形式为：111000000000 00000000010000000001，长码序列的码片速率为 4.096 Mc/s。利用这两个向量，通过 PN 码生成器可以得到长码序列如下：

10111101 01110100 00101110 10110010 10011110 11101011 01110011 11000001
00101000 11011010 01101101 11011100 10101101 00101100 01001010 00000101
11000010 10110101 01011011 01001111 10011110 11101101 00111011 10100000
00110000 10001111 10000001 10000000 11110011 10010010 01100000 00011010
11011100 01001000 00110010 01001000 10000111 11011001 01101001 10100001
11111111 01110001 11000000 00010101 11111001 00101011 00001010 10101011
01000101 00000000 01110101 01001101 10001000 01100011 01010010 00110100
00110100 00100101 10110111 11100001 01110100 10100101 00101110 11101101
11110100 00000110 11001001 00110011 10000000 11110000 01000011 10001010
01100000 01001001 1101100

这个长码序列等于在 4.096 Mc/s 速率下生成的 PN 序列。

例 10.37　选取长码序列每 64 个码片的第一个值可以实现抽取。利用例 10.36 中得到的有限的长码序列，抽取的输出 1010110010 对应了长码序列的最左列。因而，扩展的抽取输出与下面给出的充分大的长码相对应。

1010110010 0101111011 0110100011 1011100100 0101111100
0101100110 1101001001 1011010100 1101011000 1101100111
1101101100 0101001110 1101000011 0001000010 0111111000
0111001010 0010111001 1111000011 1010011110 0001010101
0110000111 1111110001 0110011011 1110111110 0001001011
1001111011 0111110101 1110001001 0101010001 0100100011
0010001100 1000100001

例 10.38　如图 10.10，每个通过 MUX 操作的多路复用符号与 64 kb/s 速率的长码 PN 码片抽取的输出结果的模 2 加运算可以得到数据扰码。只有长码序列每 64 个码片的第一个输出用在数据扰码上。通过如下形式：数据扰码 = 多路复用的 I 信道符号 ⊕ 抽取符号，前向业务 I 信道数据扰码结果如下：

I 信道数据扰码

0101001101 1010000100 1001011111 1011100100 0101000000
0110011001 0010001010 0100100100 0010100100 1101100111
0010010011 1001001110 1101111111 0010110010 0111000100
0111000101 1101000101 1111000011 1010011101 1101010101
1001000100 0000001110 0110011000 0001001110 1110001000
0101110100 0111110101 1101110110 0101010010 1000101100
0010001100 1011010001

例 10.39　考虑前向业务 Q 信道数据扰码。在例 10.35 中已经得到多路复用后的符号序列，通过如下公式：数据扰码 = 多路复用的 Q 信道符号 ⊕ 抽取符号，前向业务 Q 信道数据扰码可以经过模 2 运算得到，结果如下：

Q 信道数据扰码

0101001110	**0110001011**	1001011100	0111100100	0101111111
1001101001	1101001001	1000101011	1101100100	1110010111
1101101111	1010111110	1101000011	0010111101	0111111011
1011001010	1101000110	0000110011	0101011101	1101010101
1001000111	1111110001	0110011000	0010111110	0001001011
1001110100	0111110110	0010000110	1010010010	1000101100
0010001100	1011010001			

10.4.12　前向业务信道正交扩展

在发送之前，经过扰码的前向业务信道要与 Walsh 函数进行如下描述的正交扩展。每个前向业务 I，Q 信道与具有固定码率 4.096 Mc/s 的 Walsh 码正交扩展，以在两个 I，Q 子信道中提供正交信道化。对于 16 kb/s 速率，利用 Walsh 码 n 扩展的前向业务信道被指定 Walsh 码编号 n，$0 \leqslant n \leqslant 255$。所有 Walsh 码表是基于 Walsh64 和 Walsh64 补码的，即 $\overline{\text{Walsh64}}$。对于16 kb/s速率，至多只有 256 个信道。而我们对正交扩展的分析是基于 Walsh 码编号 $n = 72$。

例 10.40　假定 Walsh 编号 72 指定给如下的 Walsh 码：

0　　**具有 Walsh码的 Walsh码片**　　255

码编号 72	Walsh 64	$\overline{\text{Walsh 64}}$	Walsh 64	$\overline{\text{Walsh 64}}$

则具有 Walsh 码 72 的 Walsh 码片为：

00000000 11111111 00000000 11111111 00000000 11111111 00000000 11111111
11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000
00000000 11111111 00000000 11111111 00000000 11111111 00000000 11111111
11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000

例 10.41　I 信道扰码后的数据与对应着编号 72 的 Walsh 码正交扩展。利用例 10.38 中给出的 Walsh 码片 Walsh64 和 $\overline{\text{Walsh64}}$，I 信道扰码数据（只是加灰部分，即 1010000100）正交扩展如下：

I 信道正交扩展

扰码数据	扰码数据⊕**Walsh 72**
1→	11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000
0→	11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000
1→	11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000
0→	11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000
0→	00000000 11111111 00000000 11111111 00000000 11111111 00000000 11111111
0→	11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000
0→	00000000 11111111 00000000 11111111 00000000 11111111 00000000 11111111
1→	00000000 11111111 00000000 11111111 00000000 11111111 00000000 11111111
0→	00000000 11111111 00000000 11111111 00000000 11111111 00000000 11111111
0→	11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000

例 10.42　Q 信道与 Walsh 码 72 正交扩展。利用 Walsh 码 72 的 256 码片，通过 Q 信道扰码后序列（只是加灰部分，即 0110001011）与 Walsh 码片的模 2 加运算可以得到 Q 信道正交扩展。

Q 信道正交扩展

扰码数据	扰码数据 ⊕ Walsh 72
0→	00000000 11111111 00000000 11111111 00000000 11111111 00000000 11111111
1→	00000000 11111111 00000000 11111111 00000000 11111111 00000000 11111111
1→	11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000
0→	11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000
0→	00000000 11111111 00000000 11111111 00000000 11111111 00000000 11111111
0→	11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000
1→	11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000
0→	11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000
1→	11111111 00000000 11111111 00000000 11111111 00000000 11111111 00000000
1→	00000000 11111111 00000000 11111111 00000000 11111111 00000000 11111111

10.4.13　前向业务信道正交相位调制

前向业务信道经过正交扩展后要被具有固定码率 4.096 Mc/s 的导频 PN 序列正交相位调制。如同导频、同步和寻呼信道，前向业务信道也要进行如图 10.8 的正交相位扩展。导频 PN 序列由图 10.2 所示的长码生成器生成。长码满足生成器抽头多项式 $g(x)=1+x+x^2+x^{22}+x^{32}$ 所定义的线性递归。

正交扩展数据与导频码序列的模 2 加运算可以实现正交相位调制。

例 10.43　利用与例 10.12（同步信道）和例 10.22（寻呼信道）中相同的生成器抽头向量和初始存储元素向量，生成的导频码序列如下：

10111101 01110100 00101110 10110010 10011110 11101011 01110011 1100001
00101000 11011010 01101101 11011100 10101101 00101100 01001010 00000101
1100010 10110101 01011011 01001111 10011110 11101101 00111011 10100000
00110000 10001111 10000001 10000000 11110011 10010010 01100000 00011010
11011100 01001000 00110010 01001000 10000111 11011001 01101001 10100001
11111111 01110001 11000000 00010101 11111001 00101011 00001010 10101011
01000101 00000000 01110101 01001101 10001000 01100011 01010010 00110100
00110100 00100101 10110111 11100001 01110100 10100101 00101110 11101101

例 10.44　考虑前向业务 I 信道异或运算，I 信道扩展数据被导频码序列正交相位调制的过程如下公式：

I 信道正交相位调制 ＝ I 信道正交扩展数据（见例 10.41）⊕ 导频码序列（见例 10.43）

I 信道正交相位调制

01000010 01110100 11010001 10110010 01100001 11101011 10001100 11000001
11010111 11011010 10010010 11011100 01010010 00101100 10110101 00000101
00111101 10110101 10100100 01001111 01100001 11101101 11000100 10100000
11001111 10001111 01111110 10000000 00001100 10010010 10011111 00011010
11011100 10110111 00110010 10110111 10000111 00100110 01101001 01011110
00000000 01110001 00111111 00010101 00000110 00101011 11110101 10101011
01000101 11111111 01110101 10110010 10001000 10011100 01010010 11001011
00110100 11011010 10110111 00011110 01110100 01011010 00101110 00010010

例 10.45　前向业务 Q 信道异或运算，Q 信道扩展数据被导频码序列正交调制的过程如下公式：

Q 信道正交相位调制 = Q 信道正交扩展数据（见例 10.42）⊕ 导频码序列（见例 10.43）

Q 信道正交相位调制

10111101 10001011 00101110 01001101 10011110 00010100 01110011 00111110
00101000 00100101 01101101 00100011 10101101 11010011 01001010 11111010
00111101 10110101 10100100 01001111 01100001 11101101 11000100 10100000
11001111 10001111 01111110 10000000 00001100 10010010 10011111 00011010
11011100 10110111 00110010 10110111 10000111 00100110 01101001 01011110
00000000 01110001 00111111 00010101 00000110 00101011 11110101 10101011
10111010 00000000 10001010 01001101 01110111 01100011 10101101 00110100
11001011 00100101 01001000 11100001 10001011 10100101 11010001 11101101

10.4.14　前向业务信道滤波器

前向业务信道在正交相位调制之后，I，Q 脉冲要输入到如图 10.8 的 I，Q 基带滤波器进行滤波。基带滤波器具有满足如图 10.3 的限制的频率响应 $S(f)$，滤波器的归一化频率响应的通带 $0 \leq f \leq f_p$ 限制在 $\pm\delta_1$ 中，及阻带 $f \geq f_s$ 小于或等于 $-\delta_2$，以上变量的值为 $\delta_1 = 1.5$ dB，$\delta_2 = 40$ dB，$f_p = 1.96$ MHz，$f_s = 2.47$ MHz。

10.4.15　FTC 正交相移键控

如图 10.6 所示，在基带滤波之后，滤波器输出的二进制数 I，Q 根据表 10.19 映射到不同相位。

表 10.19　前向业务 I，Q 信道的信号相位点及其转换关系

相位(θ)	I		Q	
	反向不归零制		反向不归零制	
$\pi/4$	1	0	1	0
$3\pi/4$	−1	1	1	0
$-3\pi/4$	−1	1	−1	1
$-\pi/4$	1	0	−1	1

在非偏移 QPSK 中，两个基带输出在时间上重合，所以载波相位每 T_b 秒就变化一次。如图 10.8 所示，I，Q 与载波的余弦函数和正弦函数进行调幅，便可得到一个正交 QPSK 波形 $s(t)$。同步脉冲 I 被具有振幅 +1 或 −1 的余弦函数调幅，产生一个 BPSK 波形；正交相位脉冲 Q 与正弦函数调制，产生一个与余弦函数正交的 BPSK 波形。所以，两个正交 BPSK 波形的总数等于 QPSK 波形

的数目。

用 $s(t)$表示 QPSK 波形如下：

$$s(t)=I(t)\cos\omega_0 t+Q(t)\sin\omega_0 t$$
$$=\sqrt{2}\cos(\omega_0 t+\theta(t))$$

其中：$I(t)=\sqrt{2}\cos\theta(t)$和 $Q(t)=\sqrt{2}\sin\theta(t)$

通过如下对 $\theta(t)$的选择，可使得 QPSK 流 $s(t)$与 $I(t)$，$Q(t)$相一致。

1. 对于 $I(t)=1, Q(t)=1, \theta(t)=\frac{\pi}{4}, s(t)=\sqrt{2}\cos(\omega_0 t-\frac{\pi}{4})$

2. 对于 $I(t)=-1, Q(t)=1, \theta(t)=\frac{3\pi}{4}, s(t)=\sqrt{2}\cos(\omega_0 t-\frac{3\pi}{4})$

3. 对于 $I(t)=-1, Q(t)=-1, \theta(t)=-\frac{3\pi}{4}, s(t)=\sqrt{2}\cos(\omega_0 t+\frac{3\pi}{4})$

4. 对于 $I(t)=1, Q(t)=-1, \theta(t)=-\frac{\pi}{4}, s(t)=\sqrt{2}\cos(\omega_0 t+\frac{\pi}{4})$

所以，前向 W-CDMA 信道（包括导频、同步、寻呼和前向业务信道）的 $I(t)$和 $Q(t)$相位映射如表 10.19 所示，合成信号组与相位过渡如图 10.6 所示。

对 QPSK 波形绘图参数如表 10.20 所示。

表 10.20 **QPSK 波形 $s(t)$的参数表**

$\omega_0 t$	$s(t)=\sqrt{2}\cos(\omega_0 t-\theta(t))$			
	$(1,1), \theta(t)=\pi/4$	$(-1,1), \theta(t)=3\pi/4$	$(-1,-1), \theta(t)=-3\pi/4$	$(1,-1), \theta(t)=-\pi/4$
0	1	−1	−1	1
$\pi/4$	$\sqrt{2}$	0	$-\sqrt{2}$	0
$\pi/2$	1	1	−1	−1
$3\pi/4$	0	$\sqrt{2}$	0	$-\sqrt{2}$
π	−1	1	1	−1
$-3\pi/4$	$-\sqrt{2}$	0	$\sqrt{2}$	0
$-\pi/2$	−1	−1	1	1
$-\pi/4$	0	$-\sqrt{2}$	0	$\sqrt{2}$
2π	1	−1	1	1

利用表 10.20，基于 $I(t)$和 $Q(t)$的 QPSK 脉冲列 $s(t)$将在如下的例子中画出。

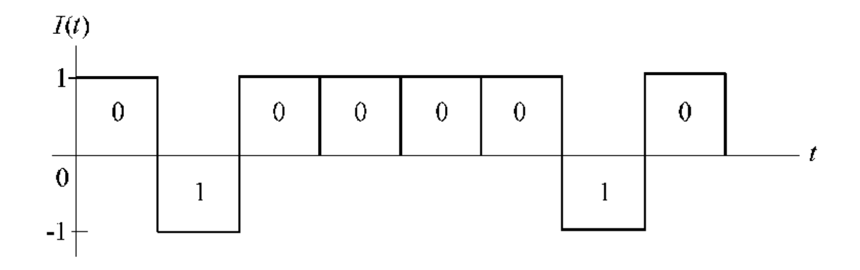
例 10.46　在节 10.4.13 的例 10.44 和例 10.45 中，我们得到 I 信道正交相位调制 $I(t)$和 Q 信道正交相位调制 $Q(t)$，即：

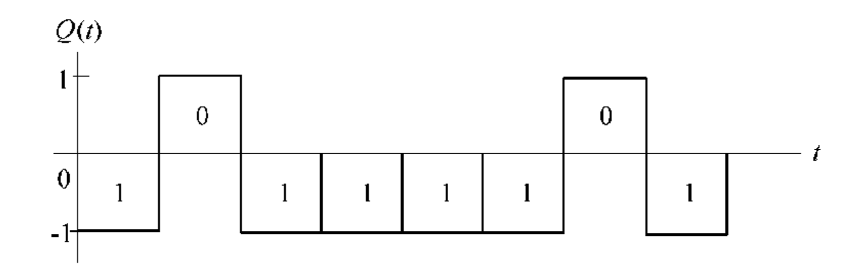
正交相位调制

I 信道：01000010 01110100 11010001 10110010 ……

Q信道：10111101 10001011 00101110 01001101 ……

只考虑加灰部分的(I,Q)脉冲对，则 QPSK 波形 $s(t)$如图 10.11 所示。





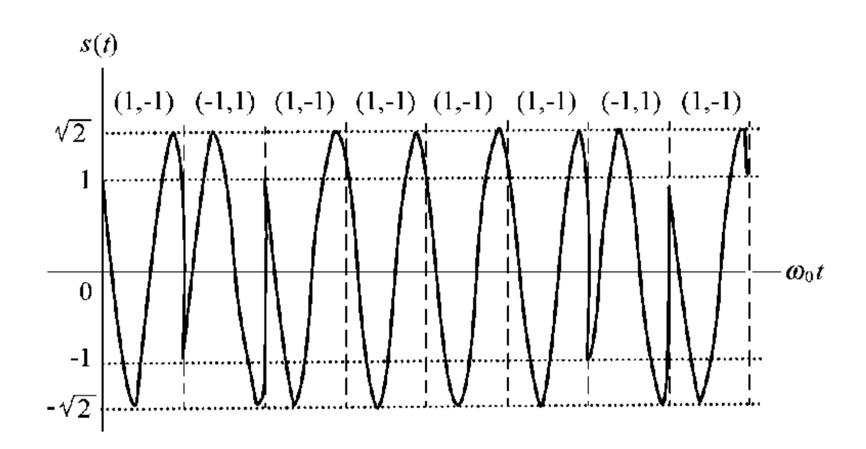


图 10.11　对应 $I(t)$ = (01000010)和 $Q(t)$ = (10111101)的 QPSK 波形 $s(t)$

术　语　表

－A－

Abbreviated alert order　缩写告警指令
Access attempt　接入尝试
Access channels　接入信道
Access channel CRC computation　接入信道循环冗余校验计算
Access channel message capsule　接入信道消息封装
Access channel preamble　接入信道前同步码
Access channel preamble length　接入信道前同步码长度
Access channel probes　接入信道探测
Access channel probe backoff range　接入信道探测补偿范围
Access channel probe sequence backoff range　接入信道探测序列补偿范围
Access channel processing　接入信道处理
Access channel slot　接入信道时隙
Access configuration sequence number　接入配置序列号
Access I channel　接入 I 信道
Access overload classes　接入过载级别
Access parameter message　接入参数消息
Access parameters　接入参数
Access parameters message　接入参数消息
Access parameters message sequence number　接入参数消息序列号
Access probe sequence　接入探测序列
Access probe　接入探测
Access Q channel　接入 Q 信道
Access redundancy　接入冗余
Acknowledgment address type　确认地址类型
Acknowledgment address type field　确认地址类型字段
Acknowledgment required field　要求确认字段
Acknowledgment required indicator field　要求确认指示符字段
Acknowledgment required indicator　要求确认指示符
Acknowledgment sequence number field　确认序列号字段
Acknowledgment sequence number　确认序列号
Acknowledgment status indicator　确认状态指示符
Acquisition　捕获

Active registration timer 激活注册定时器
Active service option number 激活服务选择号
Active service option 激活服务选择
Active set 活动集
Active set pilot 活动集导频
Adaptive differential pulse code modulation (ADPCM) 自适应差分脉冲编码调制
Address fields 地址字段
Address fields length 地址字段长度
Address type 地址类型
Alert 告警
Alert with information message 用信息消息告警
Alternate frequency assignment indicator 备用频率分配标识符
Alternative service option number 待选服务选择号
Alternative service option 待选服务选择
Analog handoff direction message 模拟切换导引消息
Analog mode operation 模拟模式操作
Analog voice channel 模拟语音信道
AND gate “与”门
Audit order 监察指令
Auth-signature procedure 鉴权签名过程
Authentication 鉴权
Authentication challenge message 鉴权查询消息
Authentication challenge response (AUTHU) 鉴权查询响应
Authentication data (AUTHR) 鉴权数据
Authentication mode (AUTH-MODE) 鉴权模式
Authentication mode field 鉴权模式字段
Authentication random challenge variable 鉴权随机查询变量
Autonomous registration 自主注册

– B –

Backoff delay 补偿时延
Base station 基站
Base station acknowledgement order 基站确认指令
Base station challenge confirmation order 基站查询确认指令
Base station challenge order 基站查询指令
Base station class 基站级别
Base station latitude 基站纬度
Base station longitude 基站经度
Baseband filter 基带滤波器

Baseband filtering 基带滤波
Binary coded decimal (BCD) 二进制编码的十进制
Binary phase shift keying (BPSK) 二进制相移健控
Bitwise complement 按位逻辑"补"
Bitwise parallel 按位逻辑"并"
Block interleaver 块交织器
Block interleaver span 块交织器跨度
Block interleaving 块交织
BPSK (binary phase shift keying) 二进制相移健控
Burst errors 突发错误

– C –

Call history parameter (COUNT) 呼叫历史参数
Call originations 呼叫源
Call processing 呼叫处理
Call termination enabled indicator 呼叫终端激活指示符
Candidate set 候选集
Capsule size 封装大小
CDMA (code division multiple access) 码分多址
CDMA channel frequency assignment CDMA 信道频率分配
CDMA channel list message CDMA 信道列表消息
CDMA channel list message sequence number CDMA 信道列表消息序列号
CDMA digital cellular system CDMA 数字蜂窝系统
CDMA mode operation CDMA 模式操作
CDMA to analog handoff CDMA 到模拟的切换
CDMA to CDMA hard handoff CDMA 到 CDMA 的切换
Cellular ratio frequency 小区无线频率
Channel assignment message 信道分配消息
Channel assignment mode 信道分配模式
Checksum 校验和
Chip time 码片时间
Cipher block chaining (CBC) mode 密码分组链接模式
Ciphertext 密文
Code channel index 编码信道索引
Code division multiple address (CDMA) 码分多址
Code excited linear prediction(CELP) 激励线性预测编码
Code repetition 码重复
Code symbol 码符号
Code symbols rate 码符号速率

Code symbol repetition 码符号重复

Coherent demodulation 相干解调

Column-by-column interleaving 列交织

Common air interface revision number(CAIREV) 公共空中接口修正号

Concatenation 串联

Configuration message 配置消息

Configuration message sequence number 配置消息序列号

Configuration parameters 配置参数

Configuration sequence number 配置序列号

Congruent 相同

Connect order 接通指令

Constraint length 约束长度

Continuous DTMF tone order 连续 DTMF 双音频指令

Conversation substate 对话子状态

Convolutional code 卷积码

Convolutional encoder 卷积编码器

Cordless telecommunications system 2 (CT-2) 无绳电话系统 2

CRC (cyclic redundancy code) 循环冗余码

CRC encoder CRC 编码器

Cryptographic key 加密密钥

Cyclic redundancy code (CRC) 循环冗余码

– D –

Data burst message 数据突发消息

Data burst randomizer 数据突发随机发生器

Data concatenation 数据串联

Data confidentiality 数据保密

Data rates 数据速率

Data scrambling 数据扰码

Data spreading 数据扩展

Daylight saving time indicator 空隙存储时间指示器

Decimal-to-binary conversion 十进制到二进制的转换

Decimated binary value 量化二进制值

Decimated sampling values 量化抽样值

Decimator 量化器

Decryption 解密

Delay operator 时延操作符

DES block cipher DES 分组密码

Digital European cordless telecommunication(DECT) 数字欧洲无绳电话

Digital signature algorithm（DSA） 数字签名算法
Dim-and-burst signaling 模糊脉冲信令
Direct sequence spreading（DSS） 直接序列扩频
Direct-sequence spread 直接序列扩频
Discrete convolution 离散卷积
Discrete logarithm problem 离散对数问题
Distance-based registration 基于距离的注册
Diversity combining 分集合并
DM algorithm DM 算法
DM scheme DM 方案
Drop timer value 撤销定时器值
DSA（ digital signature algorithm ） 数字签名算法
DSS（ direct sequence spreading ） 直接序列扩频
Dual-tone multifrequency（DTMF） 双音多频
Duty cycles 工作周期

– E –

East longitude 东经
Encoder 编码器
Encoder tail bits 编码器尾比特
Encryption 加密
Encryption call control message 加密呼叫控制消息
Encryption control message 加密控制消息
Encryption key 加密密钥
Error correction 错误校正
ESN（ electronic serial number ） 电子序列号
Excess redundancy 额外冗余
Explicit action time 显性作用时间

– F –

Feature notification message and local control order 特征通知消息和本地控制指令
Flash with information message 信息闪动消息
Foreign SID roamer 外来 SID 漫游者
Forward CDMA channel 前向 CDMA 信道
Forward information Q-channel interleaver output 前向信息 Q-信道交织器输出
Forward signaling I and Q channel frames 前向信令 I 和 Q 信道帧
Forward signaling I-channel interleaver 前向信令 I-信道交织器
Forward signaling Q-channel interleaver output 前向信令 Q-信道交织器输出
Forward traffic channels（FTC） 前向业务信道

Forward traffic channel frame offset　前向业务信道帧偏移
Forward traffic channel power control　前向业务信道功率控制
Frame　帧
Frame error rate statistics　帧差错率统计
Frame-by-frame basis　基于帧
Frame length　帧长度
Frame offset　帧偏移
Frame quality indicator　帧质量指示符
Frame structures　帧结构
Frequency division multiple access　频分多址

– G –

Generator polynomial matrix　生成多项式矩阵
GSM（ Global system for mobile communications ）　全球数字移动通信系统
GPS（ global positioning system）　全球定位系统

– H –

Hadamard matrix　哈达码矩阵
Handoff completion message　切换完成消息
Handoff direction message　切换导引消息
Handoff direction message sequence number　切换导引消息序列号
Handoff drop timer　切换撤销定时器
Handoff procedures　切换过程
Hard handoff　硬切换
Hash code　Hash 码
Hash function　Hash 函数
Hash value　Hash 值
Hexadecimal　十六进制
Hexadecimal data　十六进制数据
Home（nonroaming）　本地
Home location register/authentication center（HLR/AC）　本地位置寄存器/鉴权中心

– I –

I-channel data scrambling　I-信道数据扰码
I-channel interleaver　I-信道交织器
I-channel interleaver output　I-信道交织器输出
I-channel orthogonal modulation sequence　I-信道正交调制序列
I-channel orthogonal spreading　I-信道正交扩频
I-channel repeater output　I-信道重复器输出

I-channel repetition symbol　I-信道重复符号

I-channel symbol repetition sequence　I-信道符号重复序列

Idle handoff　空闲切换

Idle handoff procedures　空闲切换过程

Implicit action time　隐含的作用时间

Implicit registration　默认注册

Incoming call　入站呼叫

Information I channel　信息 I 信道

Information I-channel interleaver　信息 I-信道交织器

Information Q channel　信息 Q 信道

Information Q-channel interleaver　信息 Q-信道交织器

Information data rate　信息数据速率

Information rate　信息率

Initial buffer contents　初始缓冲区内容

Initial permutation（IP）　初始置换

Initial permuted sequence　初始置换后的序列

Initializing vector（IV）　初始向量

In-phase pilot PN sequence　同相导频 PN 序列

Intercept order　插入指令

Interference canceller system（ICS）　消除干扰系统

Interference canceller　干扰消除器

Interleaver array　交织器阵列

Interleaver output symbols　交织器输出符号

Interleaver span　交织器跨度

International mobile station identity（IMSI）　国际移动台标识

In-traffic system parameters messages　业务状态系统参数消息

Inverse permutation　反置换

I-pilot PN sequence　I-导频 PN 序列

– K –

Kerberos　算法名称

Key distribution center（KDC）　密钥分配中心

– L –

Latitude　纬度

Layer 2 acknowledgment fields　第 2 层确认字段

Layer 2 fields　第 2 层字段

Layer 2 procedures　第 2 层过程

Layer 3 fields　第 3 层字段

Layer 3 processing 第3层处理

Least significant bit (LSB) 最低有效位

Linear congruential generator 线性调和发生器

Local control order 本地控制指令

Lock reason 锁定原因

Lock until power-cycled order 锁定到功率循环指令

Long code 长码

Long code generator 长码发生器

Long code mask 长码掩码

Long code mask header 长码掩码头

Long code PN chips 长码 PN 码片

Long code seed 长码种子

Long code transition request order 长码转换请求指令

Long code transition response order 长码转换响应指令

Longitude 经度

– M –

Maintenance order 维护指令

Maintenance reason 维护原因

Maintenance required order 要求维护指令

Masking binary pattern 掩码二进制格式

Masking pattern 掩码格式

Maximum access channel message capsule size 最大接入信道消息封装大小

Maximum age for retention of neighbor set members 邻近集成员保持的最大寿命

Maximum length LFSR sequence 最大长度 LFSR 序列

Maximum length slot cycle 最大长度时隙周期

Maximum slot cycles 最大时隙周期

Maximum slot cycle indicator 最大时隙周期指示符

Maximum-length shift register 最大长度移位寄存器

MD2 algorithm MD2 算法

MD4 algorithm MD4 算法

MD5 algorithm MD5 算法

Mean output power level 平均输出功率级别

Mean squared error 均方误差

Message acknowledgment procedures 消息确认过程

Message address 消息地址

Message body 消息体

Message body length 消息体长度

Message digest 消息摘要

Message digest function 消息摘要函数
Message encryption 消息加密
Message encryption indicator 消息加密指示符
Message encryption mode 消息加密模式
Message encryption mode order 消息加密模式指令
Message length field 消息长度字段
Message privacy 消息保密
Message sequence acknowledgment number 消息序列确认号
Message sequence number 消息序列号
Message sequence number field 消息序列号字段
Message sequence number reset 消息序列号复位
Message transmission operation 消息发送操作
Message type 消息类型
Minimum distance 最小距离
Minimum protocol version level 最小协议版本级别
MIN1 (mobile station identification number 1) 移动台识别号 1
MIN2 (mobile station identification number 2) 移动台识别号 2
Mobile country code (MCC) 移动国家码
Mobile station 移动台
Mobile station acknowledgment order 移动台确认指令
Mobile station address 移动台地址
Mobile station control on the traffic channel state 业务信道状态上的移动台控制
Mobile station identifier (MSID) 移动台标识符
Mobile station idle state 移动台空闲状态
Mobile station initialization state 移动台初始化状态
Mobile station message transmission substate 移动台消息发送子状态
Mobile station order/message response substate 移动台指令/消息响应子状态
Mobile station origination attempt substate 移动台起始尝试子状态
Mobile station origination operation 移动台起始操作
Mobile station power-down operation 移动台功率降低操作
Mobile station registered message 移动台注册消息
Mobile station reject order 移动台拒绝指令
Mobile station transmission operation 移动台发送操作
Mobile station 移动台
Mobile station's A-key 移动台的 A 钥
Mobile station's address field 移动台的地址字段
Mobile telephone switching office (MTSO) 移动电话交换局
Modulation symbol 调制符号
Modulation symbol index 调制符号指数

Modulation symbol rate　调制符号速率
Modulo　模
Modulo-2 addition　模 2 加
Modulo square roots problem　模平方根问题
Most significant bit (MSB)　最高有效位
MSI (modulation symbol index)　调制符号指数
Multi - channel multiplexing　多信道复用
Multipath fading　多径衰落
Multiple access scheme　多址接入方案
Multiple iteractive knapsack　多交互背包
Multiple NID storage indicator　多 NID 存储标识符
Multiple SID storage indicator　多 SID 存储标识符
Multiplexed forward traffic channel　复用的前向业务信道
Multiplexing matrix for forward traffic channel　前向业务信道多路复用矩阵
Multiplicative trapdoor knapsack　乘法陷门背包
MUX　复用

– N –

NA-TDMA　北美 IS-94 方案,1990
National mobile station identify (NMSI)　国家移动台标识
Neighbor configuration　相邻配置
Neighbor configuration field　相邻配置字段
Neighbor list age　相邻列表寿命
Neighbor list message　相邻列表消息
Neighbor list message sequence number　相邻列表消息序列号
Neighbor list update order　相邻列表更新指令
Neighbor pilot PN sequence offset index　相邻导频 PN 序列偏移指示
Neighbor set　邻近集
Network identification (NID)　网络标识符
NID roamer registration indicator　NID 漫游注册标识符
Nominal transmit power offset　额定发射功率偏移
Nonlinear combiner　非线性组合器
Nonlinear filtering function　非线性滤波函数
Nonlinear function　非线性函数
Nonroaming (SID, NID) pair　本地(SID,NID)对
Nonslotted mode　非时隙模式
Non-slotted mode　非时隙模式
Normalized frequency response　归一化频率响应
Null message　空消息

Null traffic channel data　空业务信道数据

– O –

Offset index　偏移指数
Offset quadrature phase shift keying (OQPSK)　偏移正交相移键控
One-way hash function　单向 Hash 函数
Open loop power estimate　开环功率估计
OQPSK (offset quadrature phase shift keying)　偏移正交相移键控
Orders　指令
Order and message processing operation　指令和消息处理操作
Order code　指令码
Order qualification code　指令限定码
Ordered registration　受命注册
Originated call　被叫
Originating call　主叫
Origination continuation message　始呼继续消息
Origination message　发起消息
Origination operation　发起操作
Orthogonal BPSK waveforms　正交 BPSK 波形
Orthogonal channelization　正交信道化
Orthogonal modulation　正交调制
Orthogonal spreading　正交扩频
Overhead message　开销消息
Overload class　过载级别

– P –

Padded message　已填充的消息
Padding　填充
Padding length　填充长度
Page message　寻呼消息
Page record　寻呼记录
Page response message　寻呼响应消息
Page response retransmission indication　寻呼响应重传指示
Page response substate　寻呼响应子状态
Paging channel　寻呼信道
Paging channel monitoring procedures　寻呼信道监控过程
Paging channel processing　寻呼信道处理
Paging channel slot　寻呼信道时隙

Paging channel slot counter 寻呼信道时隙计数器
Paging slot 寻呼时隙
Parameter-change registration 参数改变注册
Parameter-change registration indicator 参数改变注册指示符
Parameter update confirmation order 参数更新确认指令
Parameter update order 参数更新指令
Parity check 奇偶校验
Passband 带通
PDC（ personal digital cellular ） 个人数字蜂窝
Pending message 待处理消息
Pending service option control orders 未决服务选项控制指令
Periodic power report mode indicator 周期功率报告模式标识符
Permutation function 置换函数
Permuted choice 置换选择
Permuted ESN 变序的 ESN
Persistence modifier 连续修正
Persistence modifier for access channel attempt 接入信道尝试的连续修正
Persistence modifier for registration access 注册接入的连续修正
Personal digital cellular（PDC） 个人数字蜂窝
Personal handy phone（PHP） 个人便携式电话
Personal station 移动台
Phase reference for the reverse channel 反向信道的相位参考
Phase transitions 相位过渡
Pilot and sync channel processing 导频和同步信道处理
Pilot channel 导频信道
Pilot channel acquisition substate 导频信道捕获子状态
Pilot channel number（PCN） 导频信道号
Pilot code 导频码
Pilot code generation 导频码的生成
Pilot detection threshold 导频检测门限
Pilot drop threshold 导频撤销门限
Pilot measurement request order 导频测量请求指令
Pilot offsets 导频偏移
Pilot offset index 导频偏移指数
Pilot PN offset 导频 PN 偏移
Pilot PN sequence 导频 PN 序列
Pilot PN sequence offset 导频 PN 序列偏移
Pilot PN sequence offset index 导频 PN 序列偏移指数
Pilot strength measurement message 导频强度测量消息

PIN（ personal identification number ） 个人识别号

Plaintext 明文

PMT（ Persistence modifier for access channel attempts ） 接入信道尝试的连续修正

PN chip rate PN 码片速率

PN chip sequence PN 码片序列

PN chip time PN 码片时间

PN code generator PN 码发生器

PN phase measurement PN 相位测量

PN sequence generator PN 序列发生器

Polynomial division 多项式除法

Power control 功率控制

Power control bit 功率控制比特

Power control group（PCG） 功率控制组

Power control parameter message 功率控制参数消息

Power control reporting frame count 功率控制报告帧数

Power control reporting threshold 功率控制报告门限

Power control subchannel 功率控制子信道

Power control symbol combining indicator 功率控制码元组合标识符

Power conversion 功率转换

Power increment 功率增量

Power measurement report message 功率测量报告消息

Power report delay 功率报告时延

Power-cycled order 能量周期指令

Power-down operation 关电源操作

Power-down registration 关电源注册

Power-down registration indicator 关电源注册标识符

Power-up registration 开电源注册

Power-up registration indicator 开电源注册标识符

Power-up/initialization timer 开电源/初始化定时器

Pre-output block 预输出块

Primary CDMA channel 主 CDMA 信道

Primary paging channel 主寻呼信道

Prime factorization problem 质数因式分解问题

Primitive polynomial 本原多项式

Primitive S-box functions 基本 S 盒函数

Private long code 专用长码

Private long code mask 专用长码掩码

Private long code transition request 专用长码转换请求

Propagation delay 传播时延

Protocol revision level 协议修正级

Public long code 公用长码

Public long code mask 公用长码掩码

Public long code transition request 公用长码转换请求

Public macrocellular system 公共宏蜂窝系统

Punctured multiplexing 截短复用

Punctured power control subchannel 截短功率控制子信道

Puncturing technique 截短技术

PV(n)(Persistence value for access overload classes) 接入超负荷级别的连续值

– Q –

Q-channel data scrambling Q-信道数据扰码

Q-channel interleaver Q-信道交织器

Q-channel interleaver output Q-信道交织器输出

Q-channel orthogonal modulation sequence Q-信道正交调制序列

Q-channel orthogonal spreading Q-信道正交扩频

Q-channel repeater output Q-信道重复器输出

Q-channel repetition symbol Q-信道重复符号

Q-channel symbol repetition sequence Q-信道符号重复序列

Q-channel Walsh spreading Q-信道 Walsh 扩频

Q-pilot PN sequence Q-导频 PN 序列

Quadrature modulation 正交调制

Quadrature-phase pilot PN sequence 正交相位导频 PN 序列

Quadrature phase shift keying (QPSK) 正交相移键控

Quadrature residuosity problem 积分剩余问题

– R –

Random access procedure 随机接入过程

Random challenge data (RANDU) 随机查询数据

Random challenge memory (RAND) 随机查询存储器

Random challenge value (RANDC) 随机查询值

Reciprocal polynomial 反多项式

Redundancy 冗余

Redundant data 冗余数据

Register contents 寄存器内容

Registrations 注册

Registration accepted order 接受注册指令

Registration access substate 注册接入子状态

Registration distance 注册距离

Registration message 注册消息

Registration period 注册周期

Registration persistence parameter 注册连续参数

Registration procedures 注册过程

Registration reject order 拒绝注册指令

Registration request order 请求注册指令

Registration timers 注册定时器

Registration zone 注册区域

Registration zone number field 注册区域号字段

Release order 释放指令

Release substate 释放子状态

Remaining set 剩余集

Reorder order 重排序指令

Request analog service order 请求模拟服务指令

Rescan indicator 再扫描标识符

Reset acknowledgment procedures command 复位确认过程要求

Response to overhead information operation 对开销信息操作的响应

Retransmission counter 重发计数器

Retrieve parameters message 检索参数消息

Reverse access channel 反向接入信道

Reverse access channel information bits 反向接入信道信息比特

Reverse CDMA channel 反向CDMA信道

Reverse information channel (RIC) 反向信息信道

Reverse information channel modulation parameters 反向信息信道调制参数

Reverse pilot channel 反向导频信道

Reverse signaling channel (RSC) 反向信令信道

Reverse signaling channel modulation parameters 反向信令信道调制参数

Reverse traffic channels 反向业务信道

Reverse traffic channel time offset 反向业务信道时间偏移

Roaming status 漫游状态

RS (inter-probe sequence backoff) 内部探测序列补偿

RT (probe backoff) 探测补偿

RTC (reverse traffic channel) 反向业务信道

– S –

Save-registers input parameter 存储注册输入参数

SCI (synchronized capsule indicator) 同步封装标识符

Scrambled modulation symbol sequence 扰码调制符号序列

Scrambling PN sequence 扰码PN序列

Search window size 搜索窗大小

Search window size for neighbor set 邻近集的搜索窗大小

Search window size for remaining set 剩余集的搜索窗大小

Secure hash algorithm (SHA) 安全 Hash 算法

Secure hash standard (SHS) 安全 Hash 标准

Send burst DTMF message 发送突发 DTMF 消息

Service option 服务选择

Service option control order 服务选择控制指令

Service option negotiation 服务选择协商

Service option request initialization 服务选择申请初始化

Service option request number 服务选择申请数

Service option request order 服务选择申请指令

Service option response order 服务选择响应指令

Set parameters message 设置参数消息

Shared secret data (SSD) 共享加密数据

SID roamer registration indicator SID 漫游注册指示符

SID/NID list entry timers SID/NID 列表记录定时器

Signaling channel 信令信道

Signaling code 信令码

Signaling code generation 信令码的生成

Signaling code sequence 信令码序列

Signaling constellation mapping 信号星座映射

Signaling encryption field 信令加密字段

Signaling message encryption 信令消息加密

64-ary orthogonal modulation 64 阶正交调制

Slot-cycle-index field 时隙周期指数字段

Slot cycle length 时隙周期长度

Slot length 时隙长度

Slot number 时隙号

Slotted mode 时隙模式

Slotted page message 时隙寻呼消息

Snefu 一种单向 Hash 函数

Soft handoff 软切换

SOM (start-of-message) 消息开始

Spread spectrum (SS) modulation 扩频调制

Spread spectrum signal 扩频信号

SSD-generation procedure SSD 产生过程

SSD update confirmation order SSD 更新确认指令

SSD update message SSD 更新消息

SSD update message order　SSD 更新消息指令

SSD update rejection order　SSD 更新拒绝指令

Start of message (SOM)　消息开始

Station class　基站级别

Station class mark　站级标志

Status reject order　状态拒绝指令

Status request order　状态请求指令

Stopband　截止频率

Superincreasing sequence　超递增序列

Supervisory audio tone color code　管理音频音调色彩编码

Symbol repeater　符号重复器

Symmetric block cipher algorithm　对称分组密码算法

Sync channel　同步信道

Sync channel acquisition substate　同步信道捕获子状态

Sync channel frame body　同步信道帧体

Sync channel superframe　同步信道超帧

Sync I – channel Walsh spreading　同步 I-信道 Walsh 扩频

Synchronization　同步

Synchronization message　同步消息

Synchronized capsule indicator (SCI)　同步封装标识位

Synchronized message capsule　同步消息封装

System access state　系统接入状态

System access state timer　系统接入状态定时器

System acquisition　系统捕获

System configuration　系统配置

System determination substate　系统检测子状态

System identification (SID)　系统识别号

System overhead information　系统开销消息

System parameters message　系统参数消息

System parameters message sequence number　系统参数消息序列号

– T –

TA (ack response timeout)　确认响应超时

Threshold power measurement report messages　门限功率测量报告消息

Threshold report mode indicator　门限报告模式标识符

Time-based registration timer　基于时间的注册定时器

Time-orthogonal Walsh codes　时间正交 Walsh 码

Time division multiple access (TDMA)　时分复用

Time filter　时间选通器

Time offset 时间偏移
Time randomization 时间随机化
Time randomization range 时间随机化范围
Time synchronization 时间同步
Timer 定时器
Timer-based registration 基于定时器的注册
Timer-based registration counter 基于定时器的注册计数器
Timer-based registration indicator 基于计时器的注册指示符
Timing 定时
Timing change substate 定时改变子状态
Timing information 定时消息
Traffic/access code generation 业务/接入码的生成
Traffic/access code 业务/接入码
Traffic channel frame offset 业务信道帧偏移
Traffic channel initialization substate 业务信道初始化子状态
Traffic channel processing 业务信道处理
Traffic channel registration 业务信道注册
Traffic code I sequence 业务码 I 序列
Traffic code Q sequence 业务码 Q 序列
Transmission data rate 发送数据速率
Transmission duty cycle 发送工作周期
Transmission rate 发送速率
Trapdoor knapsack problem 陷门背包问题
Truth table 真值表

– U –

Unique challenge-response procedure 惟一查询响应过程
Unlocked order 解锁指令
Unmodulated long code sequence 未调制长码序列
Unmodulated spread spectrum signal 未调制扩频信号
Unsynchronized message capsule 非同步消息封装
Update overhead information substate 更新开销信息子状态
Use action time indicator 使用激活时间标识符
Use time field 使用时间字段
UCT（ universal coordinated time ） 通用调整时间

– V –

Valid acknowledgment field 有效确认字段
Valid acknowledgment indicator field 有效确认标识符字段

Valid acknowledgment indicator　有效确认标识符
Vocoded voice　合成语音
Voice activity detection　语音激活检测
Voice activity detector (VAD)　语音激活探测器
Voice activity indication　语音激活指示
Voice channel number　语音信道号
Voice mobile station attenuation code　语音移动台衰减编码
Voice privacy　语音保密
Voice privacy field　语音保密字段
Voice privacy mode　语音保密模式

– W –

Waiting for answer substate　等待应答子状态
Waiting for mobile station answer substate　等待移动台应答子状态
Waiting for order substate　等待指令子状态
Walkie-talkie communications　无线电话通信
Walsh chip　Walsh 码片
Walsh chip period　Walsh 码片间隔
Walsh chip rate　Walsh 码片速率
Walsh code index　Walsh 码编号
Walsh code spreading　Walsh 码扩展
Walsh function　Walsh 函数
Walsh function index　Walsh 函数编号
Walsh function time alignment　Walsh 函数时间对准

– X –

X. 509　X. 509[CCIT87]最早是在 1988 年发布，以公钥密码体制和数字签名为基础

– Z –

Zones　区域
Zone-based registration　基于区域的注册
Zone list　区域列表
Zone list entry timers　区域列表记录定时器
Zone number　区域号
Zone time length　区域时间长度